

Amiga.InFo

REVISTA DEDICADA A LOS USUARIOS DE ORDENADORES AMIGA

¡¡Por fin alguien toca el Amiga en serio!!

- ✓ Número 0
- ✓ Noviembre/Diciembre 1994
- ✓ 700 ptas.

El tamaño final de la imagen es de 240x220 píxeles. Tiempo de generación: 3 horas 08 min.

¡2.5 Mb para tu AMIGA!
¡Incluye
Compilador
de C!

◆◆ DISCO 2 ◆◆

BigAnim, Chamaleon Player,
Imágenes, Módulos,
LacePointer, PCDtoIFF,
PHXAssembler, QuickGrab,
ViewTek

◆◆ DISCO 1 ◆◆

DICE C, DiskSalv,
Pager, PrintGuide,
SuperDuper

AMIGA.InFo • Nº 0



- ☐ NOTICIAS
- ☐ TUTORIAL Real 3D
- ☐ CURSO DE C
- ☐ VIDI AMIGA 12
- ☒ FRONTIER: ELITE II
- ☐ AMIGA INTERNO
- ☐ PROYECTO HARD
- ☐ SISTEMA OPERATIVO
- ☐ VIDEO TOASTER
- ☐ BITS & BAUDS
- ☐ DEMOS y JUEGOS
- ☐ EL MIDI
- ☐ AMOS

Cartas y Dudas,
Programación Teórica,
y Publicidad de los
Distribuidores Nacionales
y además

¡¡TODO LO QUE BUSCAS
SOBRE TU AMIGA EN
80 PÁGINAS Y
2 DISCOS!!



Por fin tienes en tus manos un ejemplar de AMIGA.InFo, la revista que estabas esperando ... realizada totalmente con Amiga, por gente de Amiga y para los usuarios de Amiga.

Productos Distribuidores

A-1200 Memory PC 1202-8	67
A4091 Fast SCSI-II	23
ACU: Amiga Club U.P.C.	67
AMIGA CENTER (93) 276 38 06	23
Art Department Pro 2.5	67
AT-Bus508 (A500 BUS IDE)	23

Blizzard 1220 (A1200)	40
Blizzard 1230-II (A1200)	40
Blizzard 4030 (A3000/A4000)	40
Brilliance	67

CD-ROM Mitsumi Interno	67
CD-ROM NEC 2XI SCSI-II Interno	23
CD-ROM Panasonic SCSI Interno	67
CD-ROM Toshiba Interno	40
CD-ROM Toshiba Externo	40
CIB (91) 632 27 65	48
CIMEX INFORMATICA (93) 454 70 42	57
CLUB METAL LIFE (93) 783 13 31	78
CyberStorm (A3000/A4000)	40

Desktop Dinamite Pack	67
-----------------------	----

Emplant (A3000/A4000)	23,40
Escáner Epson GT-6500	40
Escáner Epson GT-8000	40

FastLane Z3 (A3000/A4000)	23,40
---------------------------	-------

Kit CD-Rom Mitsumi + TANDEM	67
-----------------------------	----

LightWave 3.5 PAL	14
-------------------	----

M-Tec 68030/28/4 MB con MMU	67
Memory Master A1200	23
METAL SOFT (93) 783 13 31	78
Modem/Fax BZT	40
Modems USRobotics	23

OKTAGON SCSI-II	23,67
Opalvision	23

PAR: Personal Animation Recorder	14
PIXEL SOFT (979) 71 27 00	14
Photolite	67

RamBox	23
Retina 24 bits	23

The Communicator Lite	67
The Communicator MIDI	67
Tu AMIGA Ordinadors (93) 560 76 12	67

VB SOFT (93) 456 15 45	40
VideoLab 1.21	40
VIPER 030 con MMU MKII	67
V-Lab paralelo	23
V-Lab Motion	23

Warp Engine	23
-------------	----

ÍNDICE

Editorial

4

Discos del Mes

5

Instalación y uso de los más de dos Megs de programas incluidos en los discos de este mes.

Noticias

7

Las noticias y los productos más recientes aparecidos en el mercado Nacional e Internacional.

Tutorial 3D

10

Descubre el fantástico mundo de las creaciones en tres dimensiones con nuestro "viaje" al Real 3D versión 2.

Curso de C

15

Presentación del curso e instalación del compilador de C incluido en los discos del mes.

1,2 1,2 Probando

24

Este mes probamos el Vidi Amiga 12, digitalizador de vídeo en tiempo real.

Juego del Mes

27

Frontier: Elite II

Top Demos, Top Juegos

29

Nuestra lista particular de Juegos y Demos.

Distribuidores

31



¿Será este hombre el salvador del AMIGA? En la sección de Noticias encontrarás la respuesta.



¿Volveremos a ver el logotipo de Commodore en los nuevos modelos de Amiga? En la sección de Noticias encontrarás la respuesta.



Real 3D versión 2, la realidad se rendirá a tus pies con esta maravilla de la creación 3D. Este mes empezamos un tutorial.



Ejemplo de una digitalización tomada de la salida de vídeo con el VIDI 12 que hemos analizado.

Imagen creada con LightWave. Con unos resultados como éste, no es de extrañar que el Amiga esté empezando a arrasar en 3D. Dale un vistazo a la sección de vídeo.



Suscripción

32

Suscríbete hoy mismo y asegura la cómoda recepción en tu domicilio de los ejemplares de AMIGA.InFo

Amiga Interno

33

Introducción al funcionamiento del hardware interno del Amiga.

Vídeo

41

Video Toaster: La "tostadora" más potente en creación de gráficos y efectos de vídeo.

Programación Teórica

45

Explicación de los métodos utilizados actualmente para programar correctamente

Música

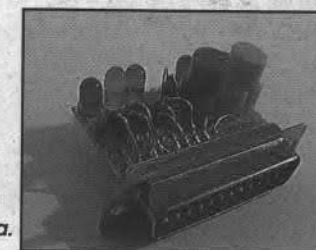
49

El interfaz MIDI, otra forma de componer música con Amiga.



Através de esta serie de artículos, desvelaremos los secretos del interfaz MIDI en la sección de Música.

Amiforo: nuestro primer montaje hard para Amiga.



Proyecto Hard

53

Amiforo: Cómo construir un periférico controlado desde el puerto paralelo.

S. Operativo

58

Introducción al sistema operativo del Amiga.

Bits & Bauds

68

Historia particular del comienzo de un aficionado en el mundo del modem.

Mercamiga

70

Compra, venta e intercambio de productos relacionados con Amiga.

Cartas y Dudas

71

Respuesta a las cartas y cuestiones planteadas por nuestros lectores.

AMOS

75

Breve introducción a la programación en AMOS Basic.



Por fin lo hemos conseguido. El primer número de AMIGA.InFo está ya en tus manos, después de pasar por muchas más. Ha sido un camino largo, pero tan apasionante que bien merece la pena volver a pasar por él para publicar el siguiente número.

La designación de éste número como el '0' es significativo y tiene sus razones. Con este ejemplar se inicia un proyecto que destaca principalmente por intentar abrir un lazo de unión entre todos los usuarios de Amiga. Sin embargo, en todo proyecto existe un riesgo y en este caso, es un riesgo que depende

directamente de ti: AMIGA.InFo sólo conseguirá sobrevivir y evolucionar si participas en ella de forma directa o indirecta. Si no es así, nada habrá servido, y todo quedará en un simple número cero.

Nosotros proporcionamos el 50% de la revista y tú, junto con los demás lectores, el otro 50%. Puedes escribirnos cartas, opiniones, dudas, redactar artículos o simplemente leerla. Lo importante es que de alguna forma nos comuniquemos tu opinión para mejorarla y adaptarla a las necesidades que compartamos.

AMIGA.InFo se compone ante todo de usuarios serios, dispuestos a demostrar que este ordenador tiene unas posibilidades de aplicación profesionales en cualquier área que se lo proponga. Por esta razón hemos incluido un conjunto de secciones que mostrarán diversos aspectos en los que el Amiga destaca y sin embargo, la información que tenemos de ellos es mínima o nula. Se nos han quedado en el tintero otras secciones y artículos que por motivos de espacio no aparecen, pero sí lo harán en los sucesivos.

En este primer número han participado 15 personas con diferentes puntos de vista y conocimientos, enriqueciendo la revista con diversidad en su contenido. A cada uno de ellos les agradezco personalmente el tiempo que le han dedicado y el reconocimiento de que sin su colaboración AMIGA.InFo no existiría. También agradecemos el interés y apoyo prestado por las empresas dedicadas a Amiga, destacando el hecho de que AMIGA.InFo tiene páginas con su publicidad de productos y, sobre todo, puede distribuirse por muchos puntos del territorio nacional.

Finalmente, no puedo resistir la tentación de escribir unas palabras acerca de la situación de Commodore. Todavía no hay una decisión oficial que resuelva de una vez por todas quién va a continuar con la tecnología Amiga, y esta situación causa cierta inquietud y tensión general que nos está perjudicando gravemente.

Commodore Reino Unido tiene grandes posibilidades de tomar las riendas, y si es así, es posible que se produzca un cambio importante en el mundo Amiga: desaparecerá Commodore del mundo Amiga y resurgirá una nueva empresa con un nuevo equipo de personas.

Pase lo que pase, el futuro a corto plazo está asegurado mientras exista el apoyo incondicional de cientos de miles de usuarios y las empresas actuales que desarrollan soft y hard. Yo encuentro la situación tan sorprendente como contradictoria: por un lado está Commodore dando sus últimos respiros y por otro los programadores y fabricantes de hard que parece que se han puesto de acuerdo en seguir desarrollando productos para Amiga. Es ahora cuando están apareciendo programas realmente potentes y profesionales que elevan al Amiga por encima de muchas otras plataformas y no hace falta que diga nombres, sólo echad un vistazo a la publicidad y la sección de noticias. En cuanto a periféricos nuevos, nunca antes habían existido tantas ampliaciones para los Amiga que ofrecen al usuario multitud de configuraciones diferentes para adaptarse a sus necesidades concretas.

Puede ser que el Amiga esté transformándose gracias a estas empresas que confían en las posibilidades increíbles de esta máquina. Ahora únicamente es necesario que el nuevo dueño de la tecnología (sea o no Commodore) vea también esas posibilidades y enfoque el Amiga al mercado profesional. Soy optimista y confío en que así será.

Sin más, espero que el contenido de la revista sea interesante y útil para todos. Saludos y hasta el siguiente número.

F. Martín



Siempre resulta difícil presentar un trabajo realizado a los demás, sin embargo, esta vez es todo un placer presentaros AMIGA.InFo. Todo empezó como una locura de dos y bueno, ahora ya es una locura de unos miles de personas. Esta revista nació con la idea de crear un canal de comunicación entre los fabricantes, los distribuidores, los programadores y por último, pero no menos importante, los usuarios de Amiga. Responde a la voluntad de atender personalmente las necesidades de todo usuario de Amiga, de hacer llegar a todos los usuarios posibles, incluso los que se encuentren en lugares remotos de España, toda la información que deseabas y que nunca encontrabas.

En muchas ocasiones el lector quiere información de las novedades o de las noticias dentro del sector, y la mayoría de veces quiere saber donde dirigirse para poder adquirir eso que tanto desea y poder comparar con ofertas reales, donde por precio, calidad, servicio,... puede encontrar el producto que más le interesa. Debido a esto hemos creído oportuno incluir toda esta información en la revista, para evitar el ya clásico peregrinaje por las distintas empresas para poder conseguir la información deseada. Desde este momento, AMIGA.InFo, te ofrecerá número a número, las ofertas y los precios de las empresas y fabricantes más representativos del sector nacional de Amiga, para que tengas la información más veraz y valiosa del mercado.

El objetivo primordial de la publicación es crear una fuente de información viva, de carácter eminentemente práctico, que se actualice en función de las necesidades de formación de los lectores. Para ello, los contenidos de la revista se dividirán en dos grandes vías: por un lado, cursos de programación y por otro, artículos y tutoriales sobre temas puntuales.

Queremos ser una revista abierta y, sobre todo, plural. Somos conscientes de que el mundo del Amiga abarca un área de inmenso tamaño, incluyendo temas muy dispares entre sí. Por ello, progresivamente, iremos tratando todos los temas que nos solicitéis, de forma que nuestros lectores puedan con el tiempo desenvolverse sin problemas en cualquier tipo de aplicaciones.

El nivel técnico de los artículos será variado ya que la mayoría de los usuarios no posee unos conocimientos de informática distribuidos uniformemente en todas las áreas existentes, sino que por el contrario, tienden a la especialización. Algunos prefieren 3D, otros la programación, otros la música, etc...

Debido a causas ajenas a nuestra voluntad nos hemos visto obligados a retrasar la salida al mercado del primer ejemplar de AMIGA.InFo (prevista para septiembre). Nuestro deseo de ofrecer información de máxima calidad y actualidad nos ha obligado a realizar algunos cambios de última hora, que han retrasado la entrada en imprenta. Ruego que disculpeis las molestias que este cambio pueda haber ocasionado, al tiempo que agradezco a todo el mundo el gran interés mostrado por ésta, vuestra revista.

Un tema importante que no quisiera dejarme es el de calidad/precio. Hasta ahora, las grandes multinacionales como IDG nos tenían acostumbrados a revistas con buen papel y repletas de color. Pero si analizabas el producto, te habían vendido 6 hojas interesantes (la publicidad) por 600 ptas. Nosotros hemos ajustado el precio al máximo y por 700 ptas te ofrecemos 80 páginas repletas de información interesante y 2 disquetes con 2 megas y medio de utilidades de todo tipo. Este precio podría reducirse significativamente si nuestra tirada fuera muy grande, pero eso depende totalmente de vosotros. De momento, podéis informar a vuestros amigos de la existencia de la revista para que la compren en el distribuidor de Amiga más cercano. Progresivamente, y con la ayuda de todos vosotros, iremos añadiendo color a las páginas, para que en el plazo más breve posible, podamos disfrutar de 80 páginas totalmente en color. Este es un buen momento para decirlos que el número 1 de AMIGA.InFo saldrá a la venta en ENERO. Será un número especial con más de 100 páginas. En este periodo de tiempo podremos crear la infraestructura necesaria para poder ofrecerte un servicio de primera (Consultas por teléfono, distribución por quioscos a nivel Nacional, nuevas secciones, etc).

Espero que AMIGA.InFo y su contenido sean del agrado e interés de todos y que consigamos afianzar el Amiga en nuestro país. Nos vemos en el siguiente número.

A. Moreno

AMIGA.InFo

APDO. DE CORREOS 166
08750 MOLINS DE REI
BARCELONA

Ninguna parte de esta revista puede ser reproducida sin autorización previa por escrito. Nosotros agradecemos cualquier colaboración pero no garantizamos su publicación. Cualquier correspondencia será considerada para su publicación si no se especifica lo contrario, y nos reservamos el derecho de editar las cartas publicadas. AMIGA.InFo reconoce todos los copyrights contenidos en este número. No obstante declinamos cualquier responsabilidad, así como cualquier garantía, que pueda ser derivada del contenido de la revista.

Dirección y Montaje: Fco. Martín y Anselmo Moreno
Redactores Artículos y Colaboradores:

Alberto Orante	Antonio Urrutia	Jacinto Javier Vaz	Santi Maspons
Anselmo Moreno	Francisco Javier Pérez	Joan Jaume López	Xavier Montero
Antoni Torrents	Ferrán García	Rafael Romero	Lluís Ribera

Si "Playboy" tiene chicas de portada, nosotros tenemos...



Aquí tenéis la primera entrega de lo que esperamos que sea una larga colección de discos de portada, cortesía de Amiga.InFo

A través de dichos discos pretendemos no sólo que dispongas de aquellas utilidades de "Public Domain" (Dominio Público, o PD), tanto de "Freeware" como "Shareware" (lee el recuadro titulado "¿Share-qué?", en esta misma página, para más información sobre PD), que harán que se te caiga la baba, ya que te pondremos en las manos lo último, lo más nuevo, lo más útil, sino que además procuraremos que te ayuden a seguir, de la manera más sencilla posible, los artículos que aparecen en cada número de la revista.

Así, no sólo incluiremos los posibles listados que surjan con los artículos de cursos de programación, para que no tengas que teclearlos (con el consiguiente ahorro de tiempo y de dolores de cabeza, al no tener que repasar continuamente el programa en busca de errores tipográficos), sino que pondremos en el disco programas que estén relacionados con los demás artículos. Naturalmente, y volviendo a referirme a los cursos de programación, también incluiremos las herramientas necesarias (léase 'compiladores') para que puedas desarrollar los diferentes programas que se proponen.

No hay ni que decir que, además de todo esto, incluiremos programas de todo tipo: "blankers" de pantalla, editores de disco y de fichero, reparadores de disco, copiadores, utilidades de impresión, de menús, del sistema operativo, del Workbench, antivirus, etc., etc., etc. Es decir, de todo, mucho y bueno.

También nos acordamos de los que les gusta la música y los gráficos, y con esto quiero decir que también incluiremos tanto programas de sonido como módulos de SoundTracker, Octamed, o cualquier otro formato interesante, e instrumentos sampleados de la más alta calidad, así como gráficos, ya sean dibujos a mano, escaneados, objetos 3D, pequeñas animaciones, logos, iconos, texturas, etc.

Entrando en materia...

Ciñendonos al tema que nos ocupa, ¿qué contienen los discos de portada de este mes? Naturalmente, material de primera, incluyendo los listados de los diferentes artículos que aparecen en este número de Amiga.InFo.

Listados

Se encuentran en el primer disco, dentro de un directorio llamado 'LISTADOS', y cada uno en un sub-directorio propio. Así, en el sub-directorio 'LISTADO/AMOS' se encuentra el ejemplo en AMOS correspondiente al artículo sobre este potente lenguaje de programación. Hablando con propiedad, el fichero está en ASCII, y, desde AMOS, hay que cargarlo con la opción 'Merge Ascii'. En el sub-directorio 'LISTADOS/MUSICA' están los dos pequeños "scripts" en ARexx, que pertenecen al tema de MIDI: Leer MIDI (Listado1) y Escribir MIDI (Listado2). En el último sub-directorio, 'LISTADO/MONTAJE -HARD', hallarás los listados en Assembler y el fichero ejecutable que sirven para probar el Amiforo.

Empezamos a lo grande, que es como se tiene que hacer, con dos discos llenitos de buen material con el que alimentar a las bestias.

Por Ferrán García.

Programas

Como se avisa en los mismos discos cuando arrancas con ellos, los ficheros se encuentran en formato comprimido, para ser más concretos, en LhA. Para los que no quieren compli-

¿Share-qué?

Existen varias formas "legales" por las que se puede designar los programas de Dominio Público, es decir, aquellos programas que están al alcance de cualquiera, dependiendo de lo que pida el autor a cambio. Algunas de éstas son:

- * **FreeWare:** el programador no pide nada a cambio, aunque en la mayoría de los casos no rechazan donativos en señal de agradecimiento, ya sean monetarios o pequeños obsequios.
- * **ShareWare:** el autor da un tiempo para probar el programa. Si después de este tiempo consideras útil el programa, debes enviarle una cantidad de dinero estipulada por aquél.
- * **CardWare:** lo único que piden es que les envíes una tarjeta postal de tu país o región, diciendo lo mucho que te ha gustado su programa.
- * **BeerWare:** "Beer" significa "cerveza"... ¿Lo coges?

Aunque parece lo más fácil coger el programa y olvidarse del programador, lo cierto es que apoyando al autor te apoyas a ti mismo, ya que ayudas a que esas mentes creativas se sientan satisfechas de su trabajo y sigan creando.

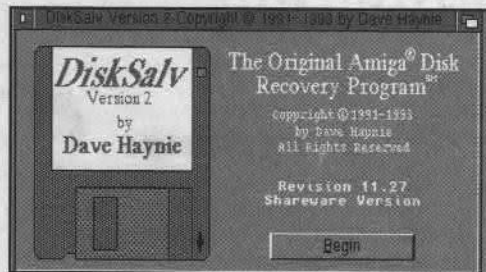
carce la existencia, tienen a su disposición un magnífico Descompresor Automático de Programas. Lo único que tenéis que hacer es introducir el disco correspondiente (da lo mismo si arrancáis con él o no), y pulsar en el ÚNICO icono que aparece al abrir el disco. Toda la operación se realiza a través del 'Installer' estándar de Commodore. Una vez ejecutado, te preguntará qué programa quieres descompactar y dónde. La descompresión se realiza sin que tengas que intervenir tú directamente.

Algunos de los programas necesitan ser instalados correctamente después de descomprimirlos. Por ejemplo, hay que asignar debidamente algunos directorios. Lo mejor que puedes hacer es leer la documentación que acompaña a cada uno de ellos, y seguir las instrucciones de instalación del propio autor del programa.

Los programas que se han incluido este mes son:

✓ DiskSalv v2

Disco 1. Esta utilidad se presenta por sí sola ya que su autor es uno de los ingenieros de Commodore más conocidos: Dave Haynie. Esta utilidad la deben tener todos los usuarios de disco duro o disquetes precavidos, ya que recupera ficheros borrados accidentalmente, desformatea, salva,



valida y repara los discos con problemas en su estructura de datos interna. Todo esto con una eficiencia elevada en la mayoría de los casos.

✓ Compilador de C DICE

Disco 1. Lo más importante no es lo que DICE sino lo que HACE. Este compilador es un sistema completo de "Editor/Compilador/Linker" de lenguaje C (estándar ANSI). Con el podéis seguir al pie de la letra el curso de C con sus listados, así como desarrollar vuestros propios programas. Quién sabe, quizás algún día veremos algún programa vuestro en estos discos...

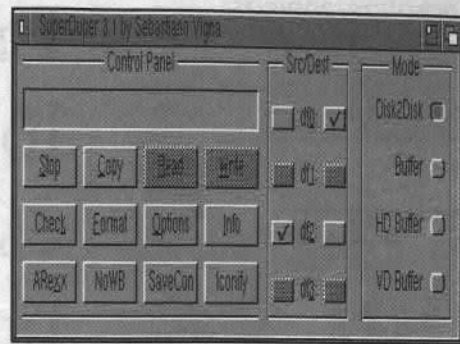
Su autor Mathew Dillon tiene ya un largo currículum en programas para Amiga y en este caso se ha creado su propio compilador rivalizando incluso con algunos comerciales.

✓ Pager

Disco 1. Pequeño programa que te ayudará a ahorrar mucho, pero que mucho, papel. Es un comando de CLI que imprime ficheros de texto aprovechando las dos caras de cada hoja.

✓ Print Guide

Disco 1. Esta utilidad, también comando de CLI, permite imprimir ficheros del tipo Amiga-Guide, eliminando los códigos propios de dicho tipo de fichero, aunque sí respeta algunos comandos, como los de principio y final de texto, o los que afecta a la apariencia del texto (negrita, itálica, etc).



✓ Super Duper v3.1

Disco 1. Este es el copiador de discos que todos los usuarios de Amiga deberían tener siempre a mano. Aparte de que es uno de los copiadores más rápidos que he visto, tiene opciones que lo hacen realmente el Nº. 1. Puedes, por ejemplo, copiar discos sin tener que pulsar una sola tecla o botón, ya que reconoce automáticamente la inserción de los discos a copiar, o te muestra una historia de los nombres de los discos copiados hasta el momento. Pero, además, y a diferencia del X-Copy, es multitarea, y su interfaz es del estilo del Intuition (Gadgets, ventanas y demás...)

✓ Big Anim

Disco 2. Como su nombre indica, ejecuta "big anims", es decir, animaciones de tamaño demasiado grande para ser cargadas totalmente en memoria. El método que utiliza es cargar en memoria las animaciones por bloques y de manera asíncrona. A medida que va mostrando los frames de la animación, simultáneamente va cargando del disco los siguientes. Las animaciones que utilizan este método generalmente se visualizan con menos velocidad que las cargadas completamente en memoria, y aunque la mayoría de nosotros no tenemos 20 mb de memoria RAM...¡sí los tenemos en el disco duro!

✓ Chamaleon Player

Disco 2. Player de módulos de SoundTracker, NoiseTracker y varios tipos más. La gran ventaja es su tamaño, ya que es muy reducido, y el hecho de que, a diferencia de otros players, se utiliza desde CLI. Además, puede quedar invisible en memoria. Es decir, lo que se dice un player sin problemas. Para saber de qué opciones dispones, teclea "champ ?" una vez lo hayas descompactado.

✓ Imágenes del mes

Disco 2. Este mes hemos incluido en el segundo disco un par de imágenes generadas por programas 3D. La primera es la cara que aparece en el tutorial de Real 3D, pero después de haber añadido unos divertidos comentarios y, como no, un poco de publicidad. La segunda imagen es una impresionante escena de la serie "Babylon-5", cuyos efectos especiales se han realizado, casi por completo, con Amiga, concretamente con "Lightwave". Como podéis ver, los resultados son espectaculares. Ah, por cierto, los que no tengáis la suerte de disponer de un A1200 o un A4000 tendréis que sudar un poco más para ver las imágenes, ya que están en modo AGA (ya sea en 256 colores o en HAM8). Pero no os asustéis, ya que podéis utilizar el programa ViewTek que viene en este mismo disco para poder visualizar cualquier formato de imagen.

✓ Lace Pointer

Disco 2. Utilidad para hacer que el puntero aparezca en modo entrelazado, es decir, más pequeño. De esta manera, puedes hacer que tu puntero se adapte en su apariencia al modo de pantalla. He incluso añade una animación del puntero cuando está en modo ocupado (busy).

✓ Módulo del mes

Disco 2. Como decía al principio del artículo, aquí tendremos de todo y para todos, y no nos olvidamos tampoco de los "modulocos", es decir, los locos por coleccionar módulos MED, ST, NT, OctaMED, etc. Aquí va el primero, "GSLINGER", un excelente módulo con mucha profusión de guitarras.

✓ PCD to IFF

Disco 2. Conversor de imágenes de formato PhotoCD a formato IFF. Muy útil para acceder a las miles de imágenes PhotoCD existentes en CDROM o incluso para ver tus propias fotos grabadas en CD con calidad de hasta 3000 dpi (Requiere CDROM, je,je)

✓ PHX Assembler v3.71

Disco 2. Macroensamblador que será utilizado para convertir en ejecutables los fuentes de la sección de Amiga Interno. Soporta y es compatible con todas las instrucciones y modos de direccionamiento de prácticamente la totalidad de los procesadores de la familia 680x0, así como de los co-procesadores matemáticos 6888x.

✓ Quick Grab v1.1

Disco 2. Capturador de pantallas compatible con los modos de pantalla AGA. Funciona también en cualquier otro modo, siempre que se tenga un WB v2.0 o superior. Puedes capturar pantallas pulsando una sola tecla.

✓ ViewTek v1.5

Disco 2. Programa que intenta mostrar en los modos gráficos nativos del ordenador aquellas imágenes que no se pueden mostrar directamente, como las imágenes de 24 bits o en formatos como GIF o JPEG. Es compatible con los nuevos modos AGA mostrando imágenes a 256 colores o en HAM8. El programa, por defecto, intenta mostrar cada imagen en el modo de pantalla que más fielmente reproduzca el original, aunque se le puede forzar a que las muestre en un modo determinado, cambiando los ToolTypes.

Bueno, esperamos que os sean de mucha utilidad todos estos programas. No dudeis en escribirnos si se os presenta algún problema con los disquetes o su contenido.

¡¡Hasta el próximo número!! ●

¡RECORDAD!: En AInFo sólo tenemos en cuenta las versiones de ROM 2.x o 3.x y con un mínimo de 1mb de memoria. Si todavía tienes un Amiga con la ROM v1.2 ó v1.3 y 1/2 mb de RAM ¿No crees que ya es hora de actualizar tu equipo?



¿CUAL ES LA SITUACIÓN DE COMMODORE?

Actualmente muchos de nosotros nos preguntamos por la situación de Commodore International, pero nadie sabe, excepto los implicados directamente en el asunto, qué es lo que está ocurriendo con la empresa que fabrica nuestros Amiga.

Los que no sepan todavía de qué estamos hablando, se sorprenderán al conocer la noticia de que Commodore Internacional (la residente en EEUU) está en proceso de liquidación tras acumular unas deudas económicas elevadas. Tras proponer la liquidación hace aproximadamente cinco meses a los tribunales de Las Bahamas, surgieron varios nombres de compradores entre los que se especulaba que estaban Hewlett Packard, Samsung o Amstrad. Sin embargo, el tiempo ha ido despejando incógnitas sobre el comprador y parece ser que la propuesta que más posibilidades tiene de éxito en la compra de C= es la subsidiaria Commodore UK en Reino Unido. ¿Commodore comprando a Commodore? Es una situación curiosa, pero posible, puesto que Commodore UK es una empresa económicamente independiente de Commodore International. Las ventas obtenidas en estos últimos años en Reino Unido han sido excelentes, debido a una política con grandes campañas de publicidad y de hecho, hasta el momento, es la única parte de Commodore que ha tenido grandes beneficios (en UK hay Amigas y CD32 hasta en la sopa). Las subsidiarias Commodore Italia, Canada y Alemania están en una situación precaria (a punto de liquidar) intentando resistir hasta que se resuelva todo este asunto.

La propuesta de compra por parte de C=UK corre a cargo de sus directores, David Pleasance (izda. foto) y Colin Proudfoot (dcha. foto), junto con dos ejecutivos senior. Uno de ellos se cree que es el director del departamento de Investigación y Desarrollo de Commodore, Lew Eggebrecht, responsable del diseño de los chips AAA. David Pleasance parece muy seguro de que sus planes van a tener éxito y así lo ha comunicado a la prensa inglesa. Ya ha anunciado que el nombre de la marca Commodore va a desaparecer y que presumiblemente se cambiará por 'Amiga



International'. De igual forma, se centralizará toda la producción y distribución de Amigas en Maidenhead (UK) y se abrirán y adaptarán las fábricas de Dundee en Escocia para producir A1200, A4000 y CD32. El hecho de que se fabriquen los Amiga en Reino Unido beneficiará la distribución en Europa, pero quizás encarezca los productos, puesto que no tendrá los mismos costes que fabricar en Filipinas.

Ahora mismo el problema más grave con el que se enfrenta Pleasance es el de surtir a todos los distribuidores europeos con nuevos stocks de A1200, A4000 y CD32. Desde hace varios meses es casi imposible encontrar Amigas, y es una situación que está perjudicando mucho a los distribuidores. Concretamente, en España, hay listas de clientes en espera a que lleguen Amigas a los distribuidores. Estos últimos están viviendo una situación asfixiante, puesto que no tienen máquinas para vender y si muchos clientes cansados de que se les diga que 'el mes o semana que viene llegarán'. Si no hay ningún cambio, para Navidades llegarán equipos. Así que un poco de paciencia...

El desarrollo de nuevos modelos de Amiga continuará en Westchester, Carolina del Norte, en EEUU. Allí todavía trabajan unos 20 ingenieros, aunque los responsables de la mayor parte del hardware y sistema operativo del Amiga, lea David Hayne, Jeff Porter y Mike Sinz se han marchado a otras empresas como SCALA y General Instruments, donde se está desarrollando el hardware para un nuevo proyecto de televisión interactiva.

Todos pensamos que los tan rumoreados chips AAA saldrán al mercado algún día, pero la última noticia es que si Commodore UK toma las riendas, no seguirá su desarrollo. Se han propuesto sacar al mercado un Amiga totalmente basado en un procesador RISC, con un rendimiento del 30% al 40% mayor que un A4000. Si esto es así, muchas cosas van a cambiar, puesto que los RISC son muy rápidos, por no decir que vuelan. El principal problema es que tendrán que adaptar el Sistema Operativo AMIGADOS para que corra en el RISC...y esto equivale a decir que pasará bastante tiempo antes de que veamos un nuevo Amiga. Sin embargo, Pleasance dice que emplearán tan sólo 12 meses (!).

Por otra parte, existen tres propuestas más de compra a tener en cuenta, una de ellas del gigante oriental SAMSUNG y dos de compañías de EEUU cuyo nombre no se ha publicado. La intención de Samsung es utilizar la tecnología Amiga para sus productos, pero no desea continuar con la línea de ordenadores. Esperemos que esta propuesta no sea aceptada...ya que en caso contrario supondría el final de Amiga. Os mantendremos informados. ●

Bienvenidos a la sección de Noticias de AInFo. Estas páginas están dedicadas a reflejar todos los sucesos relacionados con el mundo del Amiga, tanto a nivel internacional como nacional. Nuestra intención es publicar todo tipo de novedades y mostrarlas a los lectores, pero para ello es necesaria la colaboración de TODOS los distribuidores nacionales. Ponte en contacto con la redacción si eres un distribuidor de Amiga y te interesa dar a conocer tus productos nuevos a miles de lectores y además de forma gratuita.

¡Creemos un canal de información para todos!

¿CDROMs DE COMMODORE PARA AMIGA?

Utilizado ya por muchos usuarios, el CDROM se está convirtiendo rápidamente en un sistema de almacenamiento tan importante como lo pueda ser un disco duro. Prácticamente todos los PC's que se autoproclican MULTIMEDIA lo incorporan. Sin embargo, realmente pocos le sacan provecho a lo que ellos califican como MULTIMEDIA, puesto que requiere una máquina potente en el tema de gráficos y sonido utilizando tarjetas carísimas no siempre al alcance de todos.

Es aquí donde el Amiga muestra sus posibilidades, puesto que reúne todas las características para ser la plataforma MULTIMEDIA por excelencia. Y no sólo en el aspecto de hardware, sino en software, puesto que por ejemplo SCALA está a años luz por delante de cualquier otro programa que pueda existir para PC.

Commodore, sin embargo, parece que no acaba de comprender que es necesario una unidad CDROM para todos los modelos de Amiga que permita al usuario acceder a los cada vez más extendidos CD's. En principio, el A4000, que sería el modelo más beneficiado con la incorporación de un CDROM de fábrica, no tiene una unidad de la propia Commodore, sino que hay que recurrir a terceros fabricantes para conseguirla. Está claro que el A4000 es una máquina profesional y que tiene muchos modelos diferentes de CDROM's que se le pueden incorporar directamente (por ejemplo, cualquiera de PC), pero no tiene las ventajas obtenidas con la unidad instalada de fábrica.

Cuando por fin Commodore parece que cambia su planteamiento en cuanto a los CD's, responde a la demanda con el diseño de un CDROM para el A1200 que lejos de ser una solución, se convierte en un problema para muchos usuarios que deseen instalarlo. Está claramente enfocado a los usuarios que utilizan el Amiga sólo para jugar, puesto que se le ha dado preferencia a que sea totalmente compatible con la CD32. Incluso incorpora el chip AKIKO en el interior para compatibilizar, a costa de sacrificar el puerto de expansión interno del A1200. Si, si, resulta que si



conectas el CD A1200 te quedas sin tu ampliación de memoria o tu increíble aceleradora, puesto que has de extraerla para poder conectar el CD...y éste no tiene otro conector de paso... Aún así, el CD1200 incorpora un zócalo para un SIMM de 4mb

que puede solucionar el problema de memoria a algunos...pero es que ¿qué son 6 megas de memoria en un Amiga para trabajar en serio? Nada. Ha sido tan poco acertado este diseño de CDROM que aún no se ha comercializado y según se cree...nunca se comercializará, por lo menos este modelo. Y es que ahora resulta que Commodore planea sacar al mercado un A1200 con CDROM en la misma caja y un procesador M680EC30 que se llamará algo así como 'A1300'. Sin comentarios... Demos gracias a otros fabricantes por tomarse el Amiga más en serio y diseñar CDROM's en condiciones, como por ejemplo el Overdrive, comentado en la siguiente página de noticias. ●



PAGESTREAM 3.0: LA NUEVA APUESTA DE SOFTLOGIK Y AMIGA EN LA AUTOEDICIÓN

Cuando se habla de autoedición siempre aparecen los MACs, puesto que son los que tienen el software más potente en este campo. Sin embargo, según la empresa SoftLogik Publishing Corporation, se acabó el monopolio de los APPLE. Y tiene sus buenas razones para asegurarlo, gracias a una nueva versión de su programa de autoedición PageStream. Esta versión lo único que tiene en común

con la anterior, la 2.22, es el nombre, porque se han incluido cerca de 1000 nuevas opciones (!) y el 'look' es totalmente distinto. A continuación os mostramos un resumen del texto que la propia SoftLogik ha publicado. Se trata de una comparativa entre PAGESTREAM 3.0 y otros programas de autoedición de Amiga (PROPAGE 4.1) y las plataformas MAC y PC (QUARK X PRESS y PAGEMAKER 5.0):

	PAGESTREAM 3.0	PROPAGE 4.1	XPPRESS 3.3	PAGEMAKER 5.0
Número máximo de documentos abiertos	ilimitado	1	25	ilimitado
Número máx. de documentos en pantalla	ilimitado	1	1	1
Número máx. de páginas en un documento	ilimitado	9999	2000	999
Máx. ampliación vista del documento	3000%	400%	400%	800%
Núm. de tamaños de página predefinidos	15	6	5	13
Tamaño máximo de página (en cm)	508x508	147x147	147x147	106x106
Sistemas diferentes de medida	11	3	7	6
Precisión máxima	0X001 puntos	0X001"	0X001 unidX	0X001"
Capítulos y subcapítulos	X	X	X	X
Numeración de capítulos	X	X	X	X
Generación de tablas de contenido	X	X	X	X
Generación de índices	X	X	X	X
Número de tipos de fonts soportadas	3	1	2	2
Tamaños de fonts	1 to 50.000	2 to 720	2 to 720	4 to 650
Incrementos	0X001 pts	0X125 pts	0X001 pts	0X1 pts
Máxima escala horizontal de texto	1 hasta 655%	25 a 400%	5 hasta 250%	
Rotar y doblar el texto	X	X	X	X
Incrementos de rotación	0X001°	1°	0X001°	0X01°
Editor texto rotado	X	X	X	X
Filtros importa/exp. gráficos Bitmap	5/5+	5/0	7/7	7/7
Filtros imp./exp. dibujos estructurados	3/3+	2/0	7/7	7/7
Disolver dibujos en objetos	X	X	X	X
Importación de gráficos EPS	X	X	X	X
Mostrar previews de los EPS	X	X	X	X
Convertir los EPS a cualquier filtro	2/1	3/0	0/0	0/0
Salvar/Exportar las páginas como EPS	X	X	X	X
Modelos de color soportados	6	2	8	8
PANTONE	X	X	X	X
TRUMATCH	X	X	X	X
FOCALTONE	X	X	X	X
Soporte 24 bit	X	X	X	X
Impresión Postscript	X	X	X	X
Compatibilidad AMIGADOS	X 2.0-3.1	X 2.0-3.0		
Compatibilidad AGA	X	limitada		

Es evidente que PageStream resulta vencedor en muchos aspectos, pero en Amiga estamos a la espera de recibirlo y poder realizar un análisis profundo antes de calificarlo y proclamarlo el mejor como hace SoftLogik. Hasta el momento, únicamente hemos visto una copia particular, que pertenece a Tu Amiga Ordinadors. En esta copia, el manual y la presentación del paquete es muy profesional y completa, sin embargo, existe un problema que consideramos bastante grave: La versión del PageStream actual no está finalizada y muchas opciones no trabajan de forma óptima y otras no están implementadas. Parece ser que SoftLogik ha comercializado el programa en estas condiciones debido a una tremenda presión y demanda por parte de los impacientes usuarios. En breve aparecerán versiones mejoradas y gratuitas para los usuarios registrados que posean una versión anterior.●

ACELERADORAS PARA EL A4000

Recientemente han aparecido en el mercado otras dos aceleradoras para el A4000, la Warp Engine de MacroSystem Development y la G-Force 040/40 de GVP. Las dos rivalizan en el rendimiento que obtienen gracias a un procesador M68040 funcionando a 40 Mhz y convierten al A4000 en un proyectil. Las características de cada una son las siguientes:

WARP ENGINE

- Versiones: * Sin M68040 * M68040 a 28 Mhz * M68040 a 33 Mhz * M68040 a 40 Mhz

- Características:
* 4 Zócalos para la combinación de módulos SIMM de todo tipo de tamaño. (Pueden extraerse los de la placa madre del A4000 y colocarlos en la WARP obteniendo la máxima velocidad de acceso a la memoria). Hasta 128 Mb con 4 SIMMs de 32 Mb.
* Controladora SCSI-II de alta velocidad (Aprox. 3-4 Mb/Seg)

GVP 040/40

- Versiones: única con M68040 a 40 Mhz

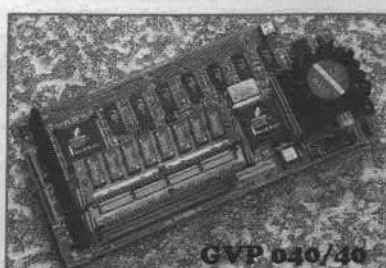
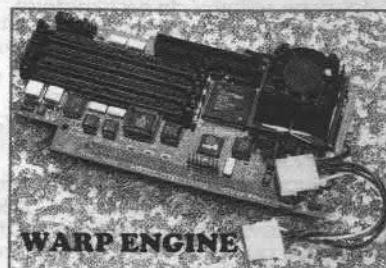
- Características:
* 2 Zócalos para módulos de memoria SIMM especiales de GVP (no sirven los de 72 pins).
* Hasta 32 Mb con SIMM de 4 Mb ó 16 Mb.

Para más información sobre la WARP ENGINE diríjase a Amiga Center (Tlf. 93 276 30 80) y sobre la GVP 040/40 al distribuidor de productos GVP en España, PixelSoft (Tlf. 979 71 27 00).●



NUEVO CDROM PARA A1200

Tiempo atrás, muchos nos sorprendimos cuando Commodore anunció que era imposible conectar un CDROM al A1200 por el puerto PCMCIA. Los problemas técnicos se debían a la intención de conseguir compatibilidad total con los juegos de la CD32. Sin embargo, encontraron la solución en el puerto de expansión interno y es allí donde se conecta el CD de Commodore y el chip AKIKO. Con la incorporación del AKIKO en el A1200, utilizado para realizar la conversión de gráficos en formato 'chunk' a formato 'planar', se consigue total compatibilidad el software de la CD32. Sin embargo, imaginaos las caras de los usuarios del A1200 que tienen ya una ampliación de memoria o una aceleradora ocupando este puerto... y quieren tener un CDROM a la vez. El CD de Commodore no es el más indicado en este caso, puesto que ocupa exclusivamente el puerto interno. La mejor solución ha aparecido con el lanzamiento por parte de la empresa francesa Archos de una unidad CDROM que se conecta al puerto PCMCIA. Y ahora le preguntamos a Commodore: ¿Pero no era imposible conectarlo al PCMCIA? Además, para más intrínquilis, resulta que este CDROM es altamente compatible con los juegos de la CD32 y Photo CD. Emplea una emulación por software del chip AKIKO tan rápida que es casi imposible distinguir al juego MicroCosm corriendo en una CD32 o en un A1200. En Amiga hemos tenido la ocasión de probarlo y es realmente impresionante ver y oír la 'intro' del MicroCosm en un A1200. Para el siguiente número incluiremos un extenso reportaje sobre él. El CDROM Overdrive (o Zappo) está disponible ya en NorSoft (Tlf. 988 24 90 46) o Tu Amiga Ordinadors (Tlf. 93 560 76 12).●



NOVEDADES EN AMIGA CENTER

- Multiface-card III de la empresa alemana BCS. Se dispone ya de la última versión de la tarjeta multipuerto reuniendo las siguientes características: 2 puertos serie alta velocidad 115.200bps y 1 puerto paralelo redireccionable para impresora. El disco de instalación incluye versión específica de PARNET para utilizarlo con esta placa.
- ENCODER para la tarjeta gráfica Retina de la empresa MacroSystem, los fabricantes de la placa gráfica. Se encuentra disponible el módulo V-code para obtener salidas video compuesto e Y/C.
- WARP ENGINE de la empresa MacroSystem. Es una tarjeta aceleradora 68040/40mhz con FAST SCSI II y ampliación máxima por SIMMs, hasta 128mb. Utiliza los SIMMs de A4000 72pin.
- Colección de CD's de dominio público que hemos recibido y las últimas actualizaciones de las siguientes colecciones:

AMINET SHARE III
17 BIT CONTINUATION
GOLD FISH
D(2 CD's) giga-pd III
Amiga Tools
CDPD III
CDPD IV
Modules_total_1

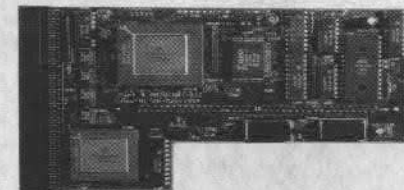
Actualización del aminet II realizado en Julio del 94.
Continuación de la librería de PD de 17bit.
Contiene la colección de Fred Fish del 1 al 1000
Ultima actualización del giga-pd (3 cd's)
Clip-arts, samples, eggs tools, vectorfonts, colorfonts...
Fred fish 761-890, imágenes 24bit, clip art, ficheros .DEM.
Fred fish 890-1000, manual c, objetos, texturas.
Colección con más de 500mb de mod's Amiga.

NOVEDADES EN TU AMIGA ORDINADORS

- Communicator Lite: La empresa holandesa Eureka ha comercializado ya la versión II del Communicator. Este aparato permite comunicar la CD32 con un Amiga o PC. Mediante él, es posible

controlar totalmente vía serie la CD32 y su CDROM, incluyendo también el módulo FMV, Photo CD's y audio. Esta versión incluye un conector para un teclado de A2000/A3000, un cable serie DB25, y un nuevo diseño externo. Disponible también una versión con MIDI. Manual traducido al castellano.

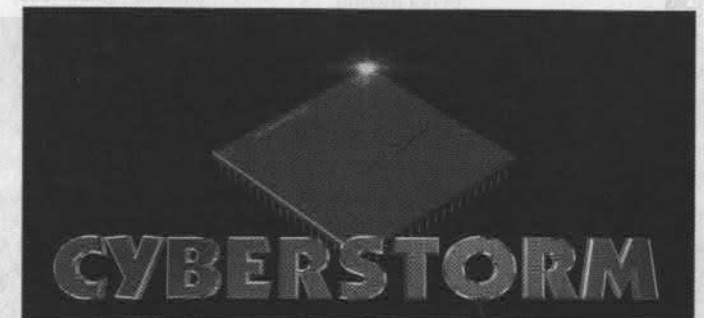
□ VIPER II: Tarjeta aceleradora para A1200 que incorpora un procesador M68030 a 40 Mhz, con MMU y un zócalo (PGA o PLCC) para coprocesador matemático y otro para un módulo SIMM de 72 pins de hasta 128 Mb.



- M-Tec: Tarjeta aceleradora para A1200 con un procesador M68030 a 28 Mhz con MMU. Incluye una memoria de 4 Mb.
- CDROM OverDrive: Unidad de CDROM para A1200 conectable al puerto PCMCIA. Para más información leer la noticia del CD1200 Zappo.●

NOVEDADES EN VBOSFT

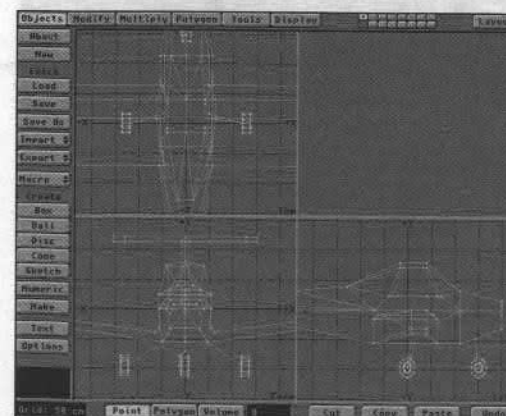
- La primera novedad es la sensacional CyberStorm, la única placa aceleradora para A4000 completamente modular. Ha sido diseñada por y para profesionales que necesitan las mejores prestaciones del mercado (en cualquier momento) y al mejor precio. Por ejemplo le permite cambiar el procesador e insertar en su lugar otro más



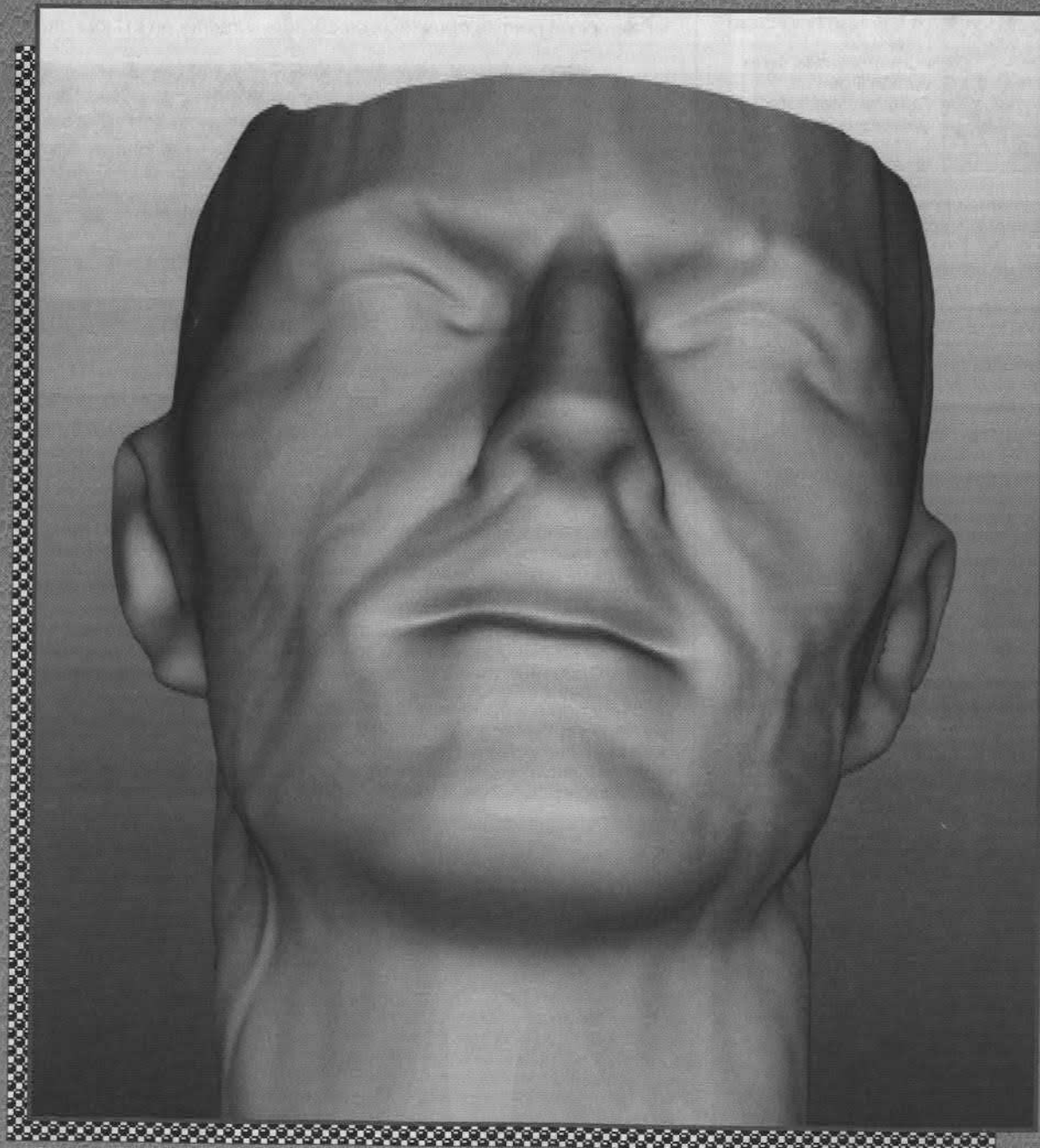
potente como de un 68040 a un 68060 o, incluso en un futuro, a un PowerPC; y todo gracias a su diseño modular que le permite montar una placa aceleradora a las necesidades de cada profesional. Esta aceleradora se compone de una Placa Madre modular y de 4 módulos (2 de ellos se incluyen siempre). La Placa Madre trabaja en modo asíncrono del resto de módulos, y es por esto que le confiere esa enorme flexibilidad. Los diferentes módulos son: Módulo CPU con 68040 ó 68060 hasta 80MHz, Tarjeta de Memoria ampliable a 128Mbytes con SIMMs de 72 pins y con transferencias de hasta 66Mbytes/s, Módulo Caché (opcional) que amplía esta memoria para acelerar aún más todos los procesos y Módulo I/O (opcional) que incluye: FastSCSI-2 (30Mbytes/s en modo 'burst' y 10Mbytes/s en modo síncrono), Ethernet BNC (Thin) y AUI (10Mbits/s, con driver SANA-2 y compatible con: Envoy, ENLAN, AmiTCP, etc.), y Serie de 115Kbits/s. El precio orientativo para la placa CyberStorm compuesta de: Placa Madre, Tarjeta de Memoria (ampliable a 128Mbytes) y Módulo de CPU con 68040 a 40MHz es de 200.000 ptas. y para la misma configuración pero con Módulo de CPU con 68060 a 50MHz es de 290.000 ptas. □ La segunda novedad es la esperada Blizzard 1220/4 para A1200, es una placa muy similar a la placa RAM-Jet (4Mbytes de memoria de 32bits incorporados, batería en tiempo real y zócalo para coprocesador matemático) pero que también incluye un procesador 68020 a 28MHz y es ampliable hasta 8 Mbytes (con un módulo opcional de 4Mbytes). Su precio orientativo es de 49.900 ptas.●

LIGHTWAVE 3.5 PAL

Quizás el programa más conocido de Amiga (sobretudo en EEUU) sea LightWave. Es más, es posible que Amiga sobreviva todavía en EEUU como el ordenador más potente en el campo del video gracias a NewTek y su increíble Video Toaster (Ver artículo de este mes en la sección de video). Ahora NewTek apuesta por el mercado de Europa y lanza una nueva versión de su programa de modelado y animación 3D: el LightWave 3.5 PAL. Este programa está creando el estándar en la industria de los programas de modelado 3D y está siendo utilizado para crear películas como Babylon 5, Jurassic Park, SeaQuest y RoboCop3. Es ya una herramienta corriente en Hollywood... y ahora todos nosotros podemos disfrutar de sus posibilidades. Bueno, todos o casi todos, puesto que para obtener resultados espectaculares se necesita un mínimo de 8 Mb de memoria, disco duro de gran capacidad, y sobretudo, un M68040 para que los renders no se conviertan en interminables. Existe una extensa colección de objetos creados por LightWave de dominio público e incluso podemos encontrar los dinosaurios de Jurassic Park en forma de objetos listos para el render y animación. Actualmente LightWave 3.5 está disponible en PixelSoft (Tlf. 979 71 27 00).●



TUTORIAL 3D



La creación de imágenes 3D mediante ordenador es un campo que siempre había estado dominado por potentes y caras estaciones gráficas. Sin embargo, actualmente, existe un ordenador que está avanzando increíblemente en este campo e incluso se permite codear con las estaciones SGI. Se trata, como no, del Amiga. En esta ocasión, Santi Maspons nos introduce en este mundo 3D a través de uno de los programas más potentes y populares: el REAL 3D Versión 2.

UN MUNDO REAL

En la plataforma AMIGA disponemos de unas excepcionales aplicaciones para trabajar en el mundo de las tres dimensiones. Tenemos herramientas para realizar el modelado de objetos, tratamiento de materiales con infinidad de propiedades, animación de escenas con el control total del tiempo y otros elementos como gases, dominio de la luz, detección de colisiones entre objetos, etc...

No pasan desapercibidos programas como LIGHTWAVE, REAL 3D, ALADDIN 4D o IMAGINE para los creadores de animaciones o de escenas estáticas. Cada uno de ellos aporta sus ventajas y depende de nuestras necesidades la elección de uno u otro. La mejor opción será la combinación de varias herramientas. Hay otras utilidades satélite de apoyo a estas maravillas de la programación como son los conversores de formatos de objetos, editores de fuentes tipográficas, proceso de los fotogramas y animaciones creados o composición de un audiovisual completo en el entorno natural multimedia AMIGA.

En esta ocasión nos fijaremos en una joya llamada REAL 3D en su versión 2. A los conocedores de la versión 1 (V1.4), hay que decir que se ha cambiado completamente el entorno de trabajo. Casi todos los programas de AMIGA son multitarea, pero en REAL 3D incluso son multitarea la totalidad de herramientas. Es decir, que mientras modelamos un objeto de la escena, podemos hacer diversos "renderings" en diferentes pantallas o ventanas simultáneamente. A continuación vamos a realizar una aventura al estilo de los antiguos exploradores para descubrir algunos de los tesoros que esconde REAL 3D.

LA EXPLORACIÓN

Nuestra nave será un Amiga 4000/040 con 6 MB de memoria real, 9 MB de memoria virtual y disco duro de 120 MB. Importante decir que REAL 3D viene acompañado de una estupenda documentación (más de 900 KB en un fichero ASCII) y de un sistema AmigaGuide más que completo.

La finalidad de este artículo es dar a conocer una forma de trabajar con REAL 3D y

promover el uso profesional del AMIGA.

Una vez instalado el programa, activamos el icono de REAL 3D y vemos como aparece una pantalla vacía en alta resolución con una única ventana reescalable titulada "Tool" y unos menús que prometen muchas opciones. Dentro de la ventana hay unos iconos con nombres como "Vis.", "Sect.", "Lev.", etc. Todas las herramientas están disponibles también en formato menú (figura 1).

Activamos el icono correspondiente a

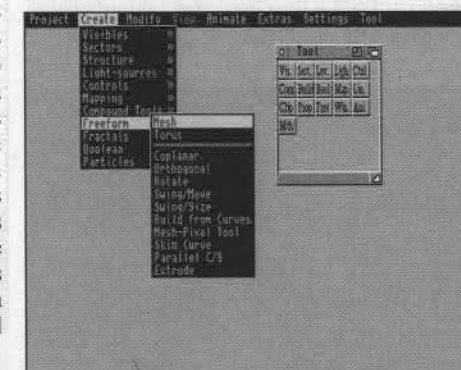


FIGURA 1.

"Win." y aparecen nuevas herramientas. La primera activa una ventana "Select" donde aparece la lista de los objetos que contendrá la escena (figura 2). La segunda abre una ventana de vista "Front View" donde se representan gráficamente los objetos. Desde los menús hay la posibilidad de abrir una ventana de vista sin bordes y a toda pantalla "Project/ Windows/ View Borderless".

Otra casilla que activamos es la "VCTR" para el control de la ventana de vista. Hemos

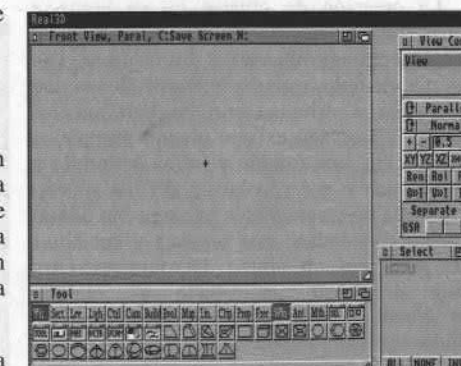


FIGURA 2.

reorganizado la pantalla moviendo las ventanas al más puro estilo Workbench. Si activamos el icono "Vis." obtenemos una representación de los objetos básicos predefinidos en REAL 3D. La figura 3 muestra todas las herramientas que poseen icono.

Para ver como funciona REAL 3D, empezaremos con la creación de un paralelepípedo o prisma. Activamos la casilla en la que hay dibujada una especie de cubo. Nos dirigimos a la ventana "Front view" y pulsamos en las dos posiciones que ocuparán los vértices.

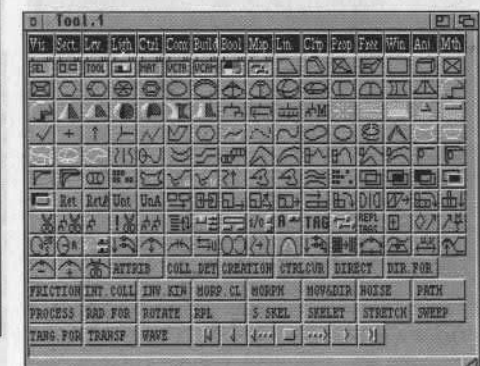


FIGURA 3

Sólo estamos observando una vista paralela de la escena. Para ver las otras vistas, pulsaremos XY, YZ o XZ en la ventana "View Con". Si en lugar de proyección paralela queremos proyección en perspectiva seleccionamos la opción "Perspect/Paralel". Los botones "+" y "-" sirven para cambiar la escala (figura 4). Nada nos impide tener varias ventanas de vista abiertas simultáneamente.

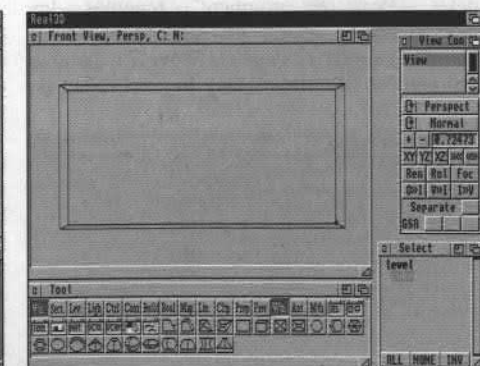
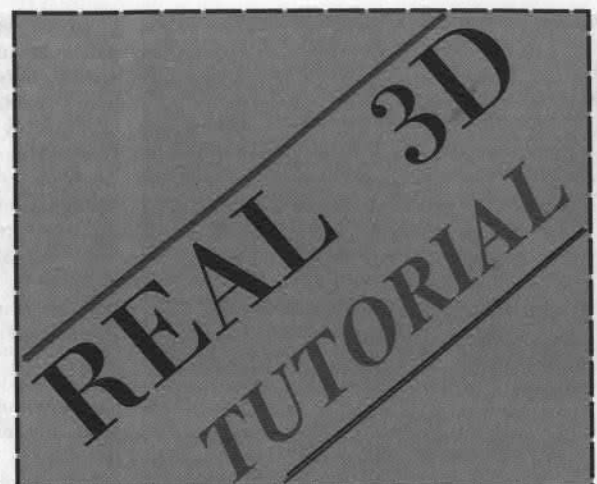


FIGURA 4.



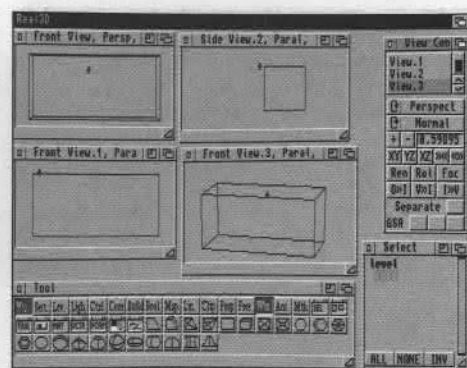


FIGURA 5.

La ventana de control de vistas contiene una lista de vistas. De esta forma podemos tener las tres proyecciones clásicas "Front", "Side", "Top" y otra en perspectiva (figura 5). La vista en perspectiva podemos obtenerla a mano moviendo las teclas de cursor en una ventana de vista o, como veremos más adelante, creando una cámara. Para modificar el objeto que acabamos de crear hay unas herramientas en la casilla "Lin." de la ventana "Tool". Podemos mover, rotar o deformar los objetos. En el menú "Modify/Linear" tenemos las herramientas en el mismo orden: mover, mover desde el centro de gravedad,

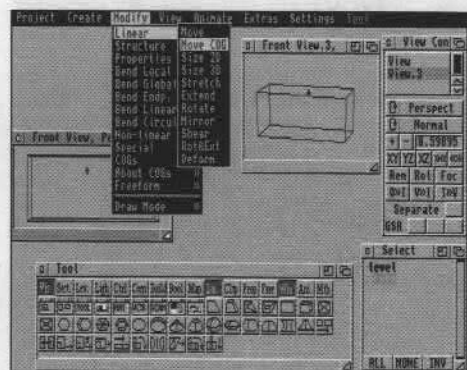


FIGURA 6.

etc... (figura 6). Para trabajar con precisión, tenemos la ventana de medidas activable desde el menú "Project/Windows/Measuring" (figura 7). Observamos que permite coordenadas cartesianas o en forma polar y sistema de coordenadas absoluto o local referido al objeto seleccionado.

Los proyectos realizados con REAL 3D incluyen la configuración del entorno de trabajo, de forma que cuando los abrimos, el entorno se configura automáticamente. En el menú "Project/environment" tenemos las operaciones de añadir un entorno al actual "Insert", guardar el entorno "Save", o pasar a otro entorno "Replace". Hay la posibilidad de

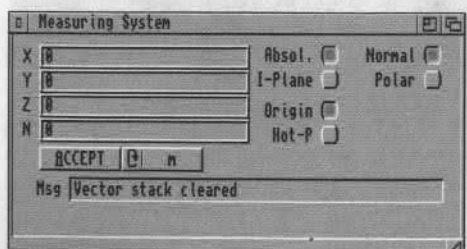


FIGURA 7.

hacer un volcado de pantalla o cambiar la paleta de colores de la pantalla de trabajo. Una opción muy interesante es la de abrir una nueva pantalla. "Open Screen" dispone de TODAS las resoluciones gráficas del AMIGA. Hacemos notar que abrir significa "abrir" y no "cambiar a", es decir, que trabajaremos simultáneamente en varias pantallas de diferentes resoluciones y número de colores. Cada pantalla puede incluir el número de ventanas y herramientas que necesitemos. Incluso podemos usar la pantalla del Workbench. En dos palabras: flexibilidad total.

Para continuar con la exploración, cambiaremos el entorno por una pantalla de super alta resolución entrelazada con 256 tonos de grises e insertaremos un objeto ya creado anteriormente. Una frágil mariposa servirá para explorar algunos conceptos de REAL 3D (figura 8). Para ver la escena desde nuestro punto de vista preferido crearemos un objeto invisible llamado cámara con el menú "View/Camera/Create Camera". La cámara es controlable mediante la ventana "ViewCam Control" que se activa en la casilla "VCTL" de la ventana de herramientas "Tool". La cámara que acabamos de crear se coloca automáticamente de forma que muestra la escena tal como la tenemos en la ventana de vista. En "ViewCam Control" cambiamos los parámetros de la cámara como el ángulo de apertura, la inclinación, las coordenadas espaciales y la posibilidad de intercambiar los

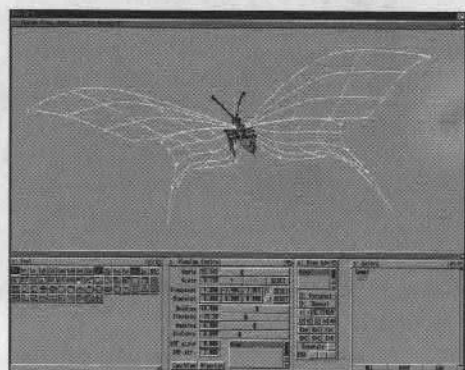


FIGURA 8.

puntos de vista de la ventana de vista y la cámara. Es posible tener una cámara para ver una escena ("Cam"), pero momentáneamente mirarla desde otro punto de vista ("ViewCam").

OBJETOS COMPLEJOS

La creación de objetos no se limita a formas básicas regulares. La mariposa del ejemplo contiene objetos de forma libre. Las alas, por ejemplo, son una rejilla de curvas que modelamos a voluntad con las herramientas "Modify". También existen objetos complejos tipo tubo con una sección y forma definibles a mano alzada o por revolución de una silueta. Otra forma de crear objetos es cargar un dibujo pequeño (entiéndase una brocha) y substituir cada pixel por un objeto determinado. Otra forma de crear un objeto es realizar operaciones entre objetos básicos: vaciamos un cilindro con un cono, interseccionamos dos esferas para crear una lente, etc...

En esta ocasión no profundizaremos en este sugestivo apartado de las tres dimensiones. REAL 3D debe mucha de su potencia a la forma en que representa los objetos de la escena. Los objetos tienen una estructura jerárquica parecida a la del sistema de ficheros con directorios y subdirectorios. Por ejemplo, podemos crear un nivel superior de objeto que incluya varias figuras básicas u otros objetos complejos. Esta representación es fundamental para comprender REAL 3D, ya que se usa continuamente para seleccionar objetos concretos o partes de la escena. No será lo mismo pegar una textura a una esfera contenida en una habitación que empapelar toda la casa. Habrá que respetar los niveles de objetos. Cuando se aplica una propiedad o movimiento a un nivel, se ve afectado todo el interior del nivel, en cambio, los niveles vecinos o superiores quedan intactos.

SÓLIDOS Y MATERIALES

Una de las características más simples que tienen los objetos es el color. Antes de crear un objeto seleccionaremos el color en la ventana de paleta de colores o modificando el color de un objeto seleccionado usando el menú "Modify/Properties/Color". Para obtener una imagen de los objetos en sólido sólo hay que

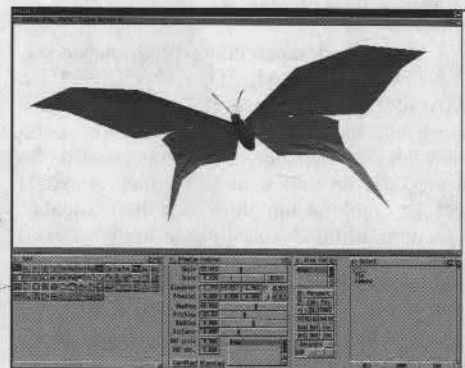


FIGURA 9.

pulsar "REN" en la ventana "View Con". Así lo hacemos, y el resultado no es el que esperamos (figura 9). Debemos configurar el proceso de "rendering" a través del menú "View/Render/Settings" (figura 10). Como no hemos colocado iluminación, usaremos el modo "LampLess" que usa un foco automático cerca de la cámara, no es muy sofisticado, pero para empezar ya nos sirve. Colocaremos el nivel de corrección de escalones "AntiAliasing" a 2. Pondremos un difuminado de fondo de blanco a negro. El resultado ha mejorado sustancialmente (figura 11).

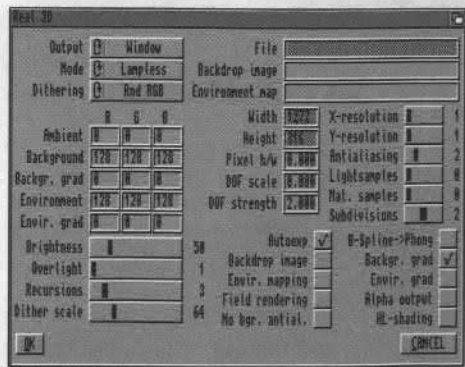


FIGURA 10.

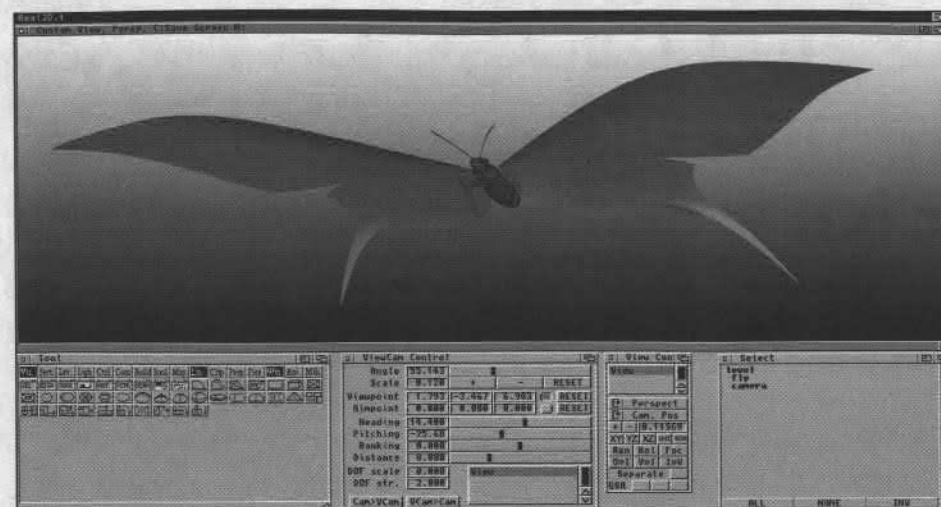


FIGURA 11.

Podemos recomendar la opción salida a fichero de 24 bits ya que el tiempo de "rendering" no se incrementa mucho y podemos procesar la imagen con otras herramientas como ArtDepartmentPro, Image Master, ImageFX o TrueBrilliance.

Aunque hemos conseguido crear objetos sólidos, no quedamos satisfechos con el resultado. Deberíamos poder hacer algo más. La solución está en los materiales. Cada objeto estará construido con un material de características controlables a través de la ventana de materiales (figura 12). El material indica como se debe tratar la luz cuando sus rayos son reflejados o refractados. De esta manera, manejamos conceptos como reflejos, transparencias, brillos y texturas.

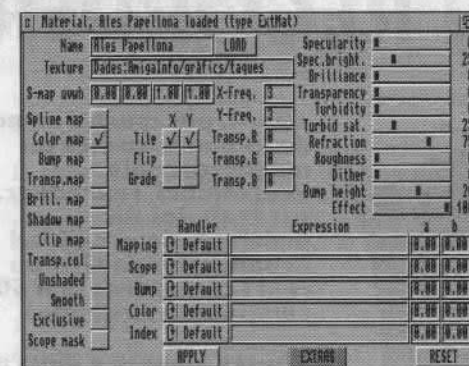


FIGURA 12.

Las texturas constan de una imagen plana creada en cualquier programa de dibujo o capturada por escáner o digitalizador de vídeo. La textura envuelve literalmente el objeto de forma plana, esférica, cilíndrica, en forma de disco o mapeado por defecto. REAL 3D representa las texturas como un objeto más con sus propiedades, tamaño y posición. También se aprovecha la textura para crear algunos efectos impresionantes. Como si de un mapa de altura se tratase, el color del dibujo que actúa de textura indica la altura de la superficie. Colores claros pueden indicar elevación de la superficie y los oscuros zonas hundidas. Con este método se genera una superficie lunar, un material rugoso o se simula la piel humana. También se puede considerar el color como indicador de transparencia de la zona (canal alfa) u otras propiedades.

En nuestra primera exploración, colocaremos unas manchas negras en las alas de la mariposa con mapeado plano, es decir, como si proyectáramos una imagen de cine sobre las alas. Cerca de la mariposa situamos

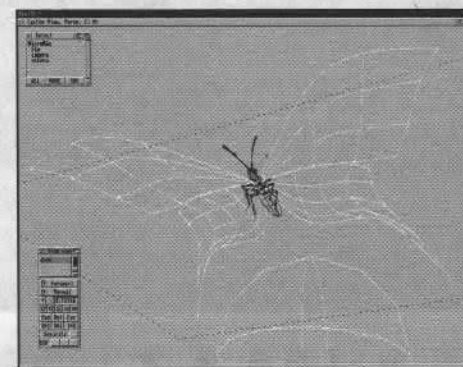


FIGURA 13.

una esfera de material brillante que actúa a modo de espejo (figuras 13).

MOVIMIENTO

Nuestros primeros pasos nos han llevado a generar la imagen estática de una mariposa, pero REAL 3D nos ofrece también la posibilidad de animarla de una forma fácil y muy elegante. No obstante, nosotros trataremos con un objeto menos bello como puede ser un cubo. Los movimientos, en REAL 3D, son tratados como otro objeto que se puede copiar, estirar, rotar o mover. Si queremos que un cubo realice una traslación, sólo debemos seleccionar el cubo en la ventana "Select", escoger la opción del menú "Animate/Create/Path" y dibujar la trayectoria en la ventana de vistas que queramos con líneas rectas o con curvas suaves. El objeto deja de ser simple y aumenta su nivel para contener un cubo y una trayectoria. Esta trayectoria puede ser modificada en su totalidad moviendo algunos puntos que la componen. Para que al mismo tiempo que realice la traslación, efectúe una rotación de un ángulo determinado, sólo debemos añadir una rotación.

Las características del movimiento, que aparecen como un objeto más en la ventana "Select", se controlan a través del menú "Modify/Properties/Animation". Hacemos notar

que si seleccionamos el cubo que acabamos de mover, la trayectoria también girará. Si queremos que el cubo haga una trayectoria fija pero que gire dentro de ella, debemos seleccionar el cubo simple original (figura 14).

Para el control del tiempo, nada tan simple como un esquema como el de la ventana "Time Line". Allí tenemos representados todos los movimientos de los objetos de forma

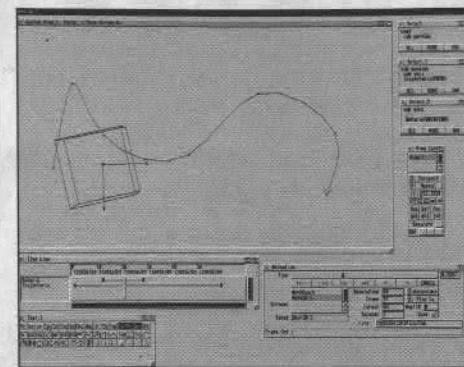


FIGURA 14.

interactiva. Los tiempos de inicio y fin de animación se varían a voluntad. Nosotros seleccionamos que la rotación se realice un rato en medio de la traslación. Cuando ya tenemos la animación preparada, controlamos el proceso de "rendering" desde la ventana "Animation" que incluye mandos análogos a los de un vídeo para visionar la animación. Si se activa la opción "WireFrame/Ray Trace" a "Ray Trace" obtenemos la imagen en sólido.

Dando nombre a los fotogramas y activando la opción "Save", guardaremos en disco cada fotograma. REAL 3D numera los fotogramas con extensión al nombre base. El número de fotogramas se escoge en "Resolution". Se puede elegir justo antes de hacer el "rendering" y olvidarnos desde un principio de cuantos fotogramas va a constar la animación. Para que todo funcione de forma automática, nos colocaremos en el primer fotograma a construir y pulsaremos el segundo mando de la ventana de animación que corresponde a un "Play" especial. REAL 3D dispone de herramientas para empaquetar fotogramas y crear un fichero de animación "Deltafile", pero cada usuario reproducirá las animaciones en su formato favorito. Si no deseamos que las ventanas de control aparezcan en medio del fotograma, usaremos una ventana de vistas sin bordes y de pantalla entera con la resolución y colores que necesitemos. Eso sí, avisando a la ventana de control de animación que debe guardar esta pantalla y no la de trabajo o la del Workbench.

El mundo REAL 3D es mucho más extenso de lo que aquí hemos apuntado. No se ha tocado el tema del potente lenguaje de programación de REAL 3D (RPL), de la metamorfosis entre objetos, soporte de tarjetas gráficas, sincronización musical, gases, colisiones, gravedad, ficheros de 32 bits con canal alpha, movimiento de estructuras articuladas y un largo etcétera. Hasta aquí este primer viaje de exploración que nos ha llevado por un mundo REAL. ●

¿Dices que haces 3D
y aún no conoces...?

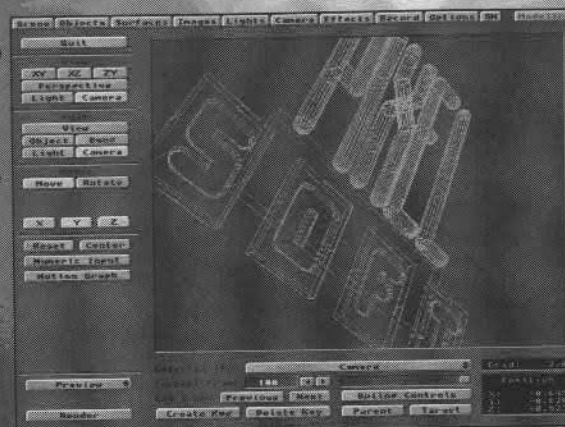


LIGHTWAVE 3.5

LightWave 3D v3.5 el más completo programa de modelado y animación 3D.

Serie de televisión de la envergadura de Babylon 5*, SeaQuest DSV o Expediente X utilizan este poderoso programa que ahora llega a España de la mano experta de PIXELSOFT.

LightWave trabaja en versión PAL y acepta resoluciones de vídeo broadcast y ultra altas como las empleadas en autoedición.



Con la garantía
y el respaldo de
PIXELSOFT

Usuarios de otros potentes programas de 3D no quieren saber nada de su antiguo software después de ver a LightWave en acción.

Disponibles también extensas librerías de objetos y módulos externos para la creación de animaciones profesionales.

NEWTEK
INCORPORATED

PIXEL
SOFT

PIXELSOFT S.L.
Felipe II, 3 BIS
34004 PALENCIA
TEL: 979.71.27.00
FAX: 979.71.28.28

*De próximo estreno en España. Disponibles capítulos piloto en videoclubs.

PERSONAL ANIMATION RECORDER

i Todavía grabando animaciones y trabajos infográficos 'cuadro a cuadro'? En más de una ocasión habrá tenido que volver a empezar el trabajo después de una larga sesión de grabación por el procedimiento del cuadro a cuadro. Y es que el procedimiento tradicional de grabación de grafismo a vídeo está supeditado a muchos imprevistos. Una simple arruga en la cinta, un drop... pueden dar al traste con horas de grabación. Eso por no hablar del desgaste de su magnetoscopio, que se ve sometido a un trabajo frenético durante este tipo de procesos.

Conozca una nueva filosofía de trabajo. Un método digital de grabación de animaciones que permitirá que Vd. vea su animación poco después de que el ordenador la haya generado, sin tener que dejarse la paciencia y el bolsillo en atenciones con los magnetoscopios y con la ventaja de la inserción real, al igual que en cine.



**NO DEJE QUE NADIE
SE 'COMA' SUS
ANIMACIONES**

Personal Animation Recorder es un dispositivo conectable a cualquier ordenador que le permitirá almacenar y reproducir vídeo digital en tiempo real. ¿Qué ventajas? No tendrá que esperar horas y horas para ver sus animaciones. Todas las copias que Vd. haga serán Masters. Inserción real. Y además, ¡cuesta mucho menos que un magnetoscopio SP!

Ahora, por mucho menos de lo que le cuesta un magnetoscopio profesional, reproduzca sus trabajos de ordenador con calidad Broadcast y en tiempo real (25 cuadros con 50 campos por segundo).

Ahorre tiempo y dinero. Dedíquese únicamente al proceso creativo. PAR se encarga del resto.

Solicite demostración.

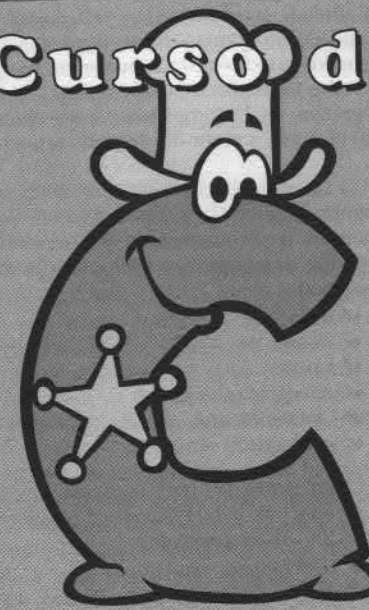


IMPORTADOR EXCLUSIVO:



PIXELSOFT S.L.
Felipe II, 3 BIS
34004 PALENCIA
TEL: 979.71.27.00
FAX: 979.71.28.28

Curso de



Comienza aquí un nuevo curso dedicado al estudio y utilización de uno de los lenguajes de programación más utilizado entre los usuarios de Amiga: el C. Con él pretendemos cubrir las necesidades de nuestros lectores y ofrecerles una idea clara y completa de este lenguaje, tanto para iniciados como para los que quieran llegar a serlo. Los menos experimentados se preguntarán qué es eso del lenguaje de programación C, y sobre todo, para qué sirve. A lo largo de este curso, A. Moreno nos dará respuesta a estas y muchas otras preguntas.

Bienvenido al excitante mundo del C, un lenguaje de programación profesional que gusta tanto a los programadores aficionados como a los comerciales. No te preocupes si los primeros listados en C te suenan a "chino". Un listado en C no es más difícil de entender que uno en Basic, cuando se conoce. Al final del curso, más de uno será capaz de mejorarlo.

Objetivo del Curso

Al comenzar cualquier actividad es esencial tener una idea clara de los objetivos que se pretenden alcanzar, de los conocimientos que se van a adquirir y del material que se va a cubrir.

Es importante señalar, antes de nada, que no se pretende iniciar un estudio de la programación a un nivel muy elevado; mas bien comenzaremos asumiendo que careces de conocimientos sobre programación.

Intentaremos introducirnos en el estudio de la programación estructurada, utilizando para ello el lenguaje C. Partiremos desde la base más elemental, para que incluso quien no tenga ni la más remota idea de lo que es esto pueda seguirnos.

El objetivo global de este curso es proporcionarte una base completa (en todos los aspectos), de la programación en C para sacarle el máximo partido al Amiga. Objetivo suficientemente amplio y completo, como para hacer un curso de, por lo menos, un año de duración.

¿A quién va dirigido?

Este curso puede ser usado por una gran variedad de lectores, ya que cubre dos necesidades. Es un texto de referencia para el experimentado programador, o bien una obra de estudio personal para el nuevo programador interesado o implicado en la construcción de software para Amiga.

Se adopta el lenguaje C para fines generales y, por tanto, será útil tanto para estudiantes como para profesionales. Sin embargo, resulta eficaz usarlo junto con otros cursos sobre temas como: Sistemas Operativos, diseño de Software, lenguaje ensamblador, estructuras de datos, etc.

Podrán beneficiarse con este curso:

Programadores principiantes que han tenido alguna experiencia en AmigaBASIC, AMOS, Blitz Basic o GFA-BASIC y que ahora quieren aprender C.

Programadores experimentados que saben BASIC, Pascal, Modula-2 o C, y que ahora lo quieren aplicar al Amiga.

Programadores de Amiga que quieren explicaciones y ejemplos en castellano del control de memoria, control de pantalla y teclado, ficheros, entrada y salida, gráficos, sonido, etc.

El programador de Sistemas que necesite utilizar C para trabajar en un medio UNIX, encontrará una explicación completa sobre esto. Explicaremos cómo procesar archivos y proporcionaremos los elementos

necesarios del lenguaje para programar sistemas. Analizaremos con detenimiento varias bibliotecas estándar de funciones de entrada y salida.

No se da por supuesto en ningún momento que seas un programador experimentado. Así, se explican ideas de programación junto con el propio lenguaje. El curso ha sido diseñado con la idea de hacerlo inteligible para todo el mundo, tanto para programadores principiantes como para experimentados. Se pretende que este curso sea una introducción amena y sencilla al fascinante mundo de la programación del Amiga.

En definitiva, va dirigido a todos aquellos deseosos de aprender con rapidez el lenguaje C en sus múltiples variantes (ANSI C, UNIX C y C bajo AmigaDOS).

CURSOS DE PROGRAMACIÓN DE AMIGA.InFo

SERVICIO GRATUITO DE

CONSULTAS

Si no has comprendido algún ejemplo, si tienes problemas con tu compilador o si dudas del empleo de alguna expresión, el equipo que AMIGA.InFo pone a tu servicio está siempre dispuesto a dar respuesta a tus preguntas. Además, si lo deseas, abrirán una ficha con los datos que obtengan de tus consultas y te guiarán como si fueran tu tutor.

Es muy probable que el gran volumen de lectores nos haga imposible mantener una correspondencia personalizada, pero aún así, nos agradecería que quienes sigan los cursos nos escriban contándonos sus progresos o las dificultades que encuentran. Estas cartas nos permitirán ir adaptando las explicaciones a un nivel que satisfaga a todos y permita que nadie se quede "descolgado".

No estudias solo. Mientras duren los cursos puedes consultar a un equipo de especialistas que dará respuesta a tus dudas de comprensión, de programación, sobre manuales o documentación...

Puedes aclarar tus dudas por carta a: **AMIGA.InFo (Ref: Consultas-C)**
Apartado de Correos 166
08750 Molins de Rei (Barcelona)

¿Qué contiene?

Es casi imposible describir, en un único curso, el lenguaje de programación C. En la práctica hay dos tipos de textos sobre C:

a) Tutoriales sobre programación en C.

Los libros que siguen el estilo tutorial están repletos de ejemplos escritos en C. Puedes teclearlos y ejecutarlos perfectamente, e incluso te aconsejan que modifiques el ejemplo y aprendas de los resultados.

b) Referencias del C.

Los libros que siguen el estilo de referencias suelen ser una descripción comprensiva de todas las llamadas estándar posibles.

Para este curso he optado por hacer una rápida, pero extensa, referencia sobre el lenguaje C para pasar posteriormente a hacer un tutorial sobre C para Amiga. Los motivos son evidentes. Existe gran cantidad de libros para aprender C, desde un nivel muy básico hasta un nivel experto; sin embargo, no existe casi ningún libro, y sobre todo en castellano, para aprender a programar en C nuestro Amiga. Por lo tanto, la parte encargada de explicar el estándar ANSI-C tendrá menor número de ejemplos prácticos, en cambio, la parte de aplicación del C a Amiga estará repleta de ejemplos. Creo que verás más interesante un ejemplo sobre Amiga que un clásico "Hello World" de C; aunque, por supuesto, lo llevará ya que es un curso para aprender C desde cero.

El curso consta de cuatro partes claramente diferenciadas:

□ La parte I ofrece una completa descripción -con numerosos programas y ejercicios- del lenguaje de programación C,

constituyendo un verdadero manual del mismo. Los diferentes artículos harán que pases de los conceptos simples a las estructuras de datos complejas y que todos los aspectos del lenguaje C se vayan cubriendo de forma progresiva.

Su objetivo es enseñar al lector que se enfrenta por primera vez con este lenguaje los conceptos fundamentales del mismo, para construir los cimientos necesarios para poder abordar con éxito las partes restantes del curso. Así mismo, también se ha pensado para que sirva de ayuda para todos aquellos programadores experimentados que deseen consultar algún aspecto del lenguaje de programación C. Por ejemplo, incluye:

- ✓ Introducción de conceptos básicos necesarios para todo programador.
- ✓ Uso de los editores de texto.
- ✓ Recomendaciones para escribir código legible.
- ✓ Introducción a los compiladores y otras herramientas necesarias.
- ✓ Uso y estudio de los cuatro compiladores de C más usados de Amiga.
- ✓ Justificación de la buena programación.
- ✓ Completa descripción del lenguaje de programación C.
- ✓ Comentarios y avisos sobre posibles errores de programación.
- ✓ Rutinas de Entrada/Salida para sus programas en C.
- ✓ y muchísimas cosas más...

Incluye explicaciones completas y comprensibles de las facetas de C de uso más común. Se tratan la mayoría de las recomendaciones del estándar ANSI propuesto actualmente. Además, el curso presenta una aproximación actual a la programación, insistiendo en la importancia de la claridad,

legibilidad, modularidad y eficiencia en el diseño de programas. Se te presentan de esta forma los principios de la buena programación, así como las reglas específicas del C. Aparecen programas en C completos y para Amiga por todo el curso, a partir del segundo artículo.

□ La parte II cubre técnicas avanzadas de programación en C, profundizando en aspectos concretos de programación y ofreciendo una colección de algoritmos básicos en problemas relacionados con:

- ✓ Pilas
- ✓ Colas
- ✓ Listas
- ✓ Árboles
- ✓ Ordenamiento
- ✓ y Búsqueda

□ La parte III trata aspectos específicos del AmigaDOS y del entorno de programación del SAS/C. Estos incluyen:

- ✓ "Debuggear" en Amiga.
- ✓ Explicación completa de los aspectos más relevantes del sistema.
- ✓ Multitud de ejemplos listos para ser usados en tus programas.
- ✓ y un largo etcétera.

Se incluyen muchos ejemplos como parte principal del texto. Estos incluyen a su vez numerosos ejemplos de programación de complejidad variable, así como ejemplos ilustrativos de las características especiales del sistema operativo del Amiga.

Por último:

□ La parte IV está formada por una serie de apéndices. Estos apéndices suministran una serie de tablas, reglas de sintaxis y símbolos de gran utilidad para programar en C. Constituyen una referencia concisa e indispensable que los

programadores de Amiga podrán usar como referencia. Entre otros, contiene los siguientes apéndices:

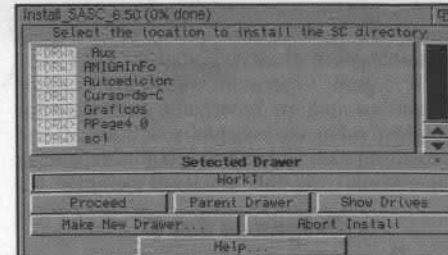
- ✓ Librerías de C.
- ✓ Librerías del Amiga.
- ✓ Códigos ASCII.
- ✓ Códigos de Escape.
- ✓ Mensajes de error del SAS/C.
- ✓ Soluciones a los problemas propuestos.
- ✓ etc.

Al final de cada artículo aparecen conjuntos de ejercicios y cuestiones de repaso que te permitirán probar la asimilación del material presentado. Deberías resolver el mayor número posible de estos problemas de cara a conseguir un aumento de la confianza en tí mismo y estimular de esta forma tu interés por el tema. Recuerdo que para la programación se necesita una cierta habilidad, de forma semejante a como se necesita para tocar un instrumento o escribir un libro, y esta habilidad no se adquiere simplemente leyendo este curso.

¿Qué se necesita?

A fin de obtener el máximo rendimiento de este curso, necesitarás lo siguiente:

□ Uno de los muchos compiladores de C existentes para Amiga. La lista de compiladores que hay en el mercado es muy amplia, pero nos servirá cualquiera de los siguientes: Lattice C, SAS/C, GCC, DICE, Aztec C, North C o cualquier otro de los existentes en las colecciones de Dominio Público. Debido a que no todo el mundo tiene un compilador de C en casa, he creído oportuno incluir en el disco de portada, la versión "Shareware" de DICE. Te recomiendo encarecidamente que, si quieres convertirte en todo un programador profesional de C, te



SAS C es, sin lugar a dudas, el mejor compilador de C para Amiga. A lo largo del curso analizaremos a fondo éste y otros compiladores.



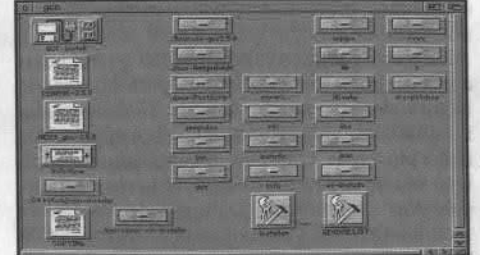
El compilador Aztec C será válido para seguir el Curso de C, aunque te recomiendo que cambies a: DICE, GCC, o mejor aún, a SAS/C.

compres uno de los compiladores de C mencionados anteriormente.

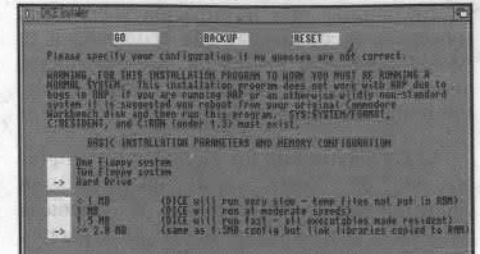
□ Por supuesto, un Amiga, con al menos:

	RAM	Disco Duro	Discos
GCC	3-4 Megas	30 Megas	---
DICE	2 Megas	Recomendable	3
SASC	2-3 Megas	10 Megas	6
Aztec C	2 Megas	Recomendable	4

□ Una o dos unidades de disco, o una unidad de disco y un disco duro.



GNU CC es un compilador cruzado de ámbito universitario. Puede crear un ejecutable para 680X0, PC, VAX, estaciones SUN, etc.



Instalación de la versión registrada de DICE. Incluye el set completo de "Includes" de Commodore, el código fuente de la c.lib, soporta operaciones en coma flotante, etc.

□ Un televisor o monitor en color, ya que algunos ejemplos llevan gráficos o texto en color (aunque en la mayor parte del curso no se necesite el color).

□ Una impresora (opcional). Varios de los ejemplos requieren el uso de la impresora.

□ Workbench 2.0 o preferiblemente 3.0. He centrado el curso en las versiones más recientes del sistema operativo, ya que todavía hay muy poca documentación, y ninguna en

DICE: Compilador de C

Nombre: DICE

Autor: Matthew Dillon
891 Regal Rd.
Berkeley, Ca. 94708
USA

Descripción:

DICE son las siglas de "Dillon's Integrated C Environment". Actualmente está considerado como el compilador de C, Shareware, más potente para Amiga. El calificativo de Shareware es debido a que su distribución es gratuita, aunque es recomendable el envío de una bonificación al autor. Por un módico precio, recibirás la versión completa en casa. Lleva el compilador y el enlazador o montador (en inglés, linker), pero además, incluye un excelente editor de textos con opciones muy potentes para conseguir un entorno con todo lo necesario, para poder empezar a programar en C. Está acompañado de una amplia documentación, eso sí, en inglés pero si no desear gastarte mucho dinero en la adquisición de un compilador de C para seguir este curso, la versión registrada de DICE es la mejor opción.

Principales características de DICE:

□ Es un 95% compatible con el ANSI C, incluye casi todas sus funciones estándar.

□ DME, el editor de textos que acompaña a DICE, permite hacer una rápida referencia en cualquier momento. Simplemente hay que colocarse sobre el nombre de una función, pulsar una tecla, y al momento dispondremos del manual de esa función en una ventana aparte.

□ Soporta todos los WorkBench. Dispone de un módulo estándar de arranque compatible con todas las versiones existentes del Workbench, es decir, cuando un programa es ejecutado desde el Workbench, se llama a una función diferente de entrada: la wmain(wbmsg). De lo contrario usaríamos la librería c.lib y al detectar la ejecución de nuestro programa desde el Workbench, retornaría inmediatamente un -1 y abortaría la ejecución del programa.

□ DICE es bastante rápido. Cuando hacemos residentes los diferentes módulos que forman el compilador, conseguimos que la

compilación casi "vuela". Gran parte de esta velocidad se consigue gracias a cargar de golpe ficheros enteros en la memoria. Esto presenta un pequeño problema de memoria, pero en general, el uso que hace DICE de la memoria es, más o menos, el mismo que hace Aztec C y Lattice (primeras versiones del SAS/C).

□ El código que genera es compacto y bastante reducido en general. Los tamaños de los programas más pequeños pueden rondar los:

```
_main() { } 448 bytes, no utiliza stdio/fd.  
main() { } 2736 bytes, utiliza algo de stdio.  
main() { algún printf } 5492 bytes, utiliza gran parte de stdio.
```

□ El preprocesador es capaz de tratar macros de cualquier tamaño. No tiene límite de símbolos, y además, permite manejar hasta 64 niveles de "includes" y recursiones en las macros.

□ Soporta la creación de ejecutables residentes sin que el programador tenga que realizar trabajo alguno adicional.

□ Soporta completamente las versiones del sistema operativo 1.3 y 2.0, y con un fácil mecanismo de extensiones al final de los

(Incluido en el disco de Portada)

nombres de los ficheros, permite desarrollar programas para varias versiones de sistema operativo simultáneamente.

□ Incluye el código fuente de las diferentes librerías: c.lib, m.lib, auto.lib, etc. para poder corregir los bugs que se detecten y así compilar de nuevo la librería ya corregida. Además, incluye el código fuente de casi todas las utilidades que lo acompañan, para que el nuevo programador disponga de ejemplos de gran calidad de donde aprender técnicas de programación en C.

□ Posee una extensa documentación en inglés sobre cómo instalar el compilador, su utilización, el editor, las funciones estándar de C, las librerías suministradas, cómo realizar librerías propias, cómo generar código para ROMs y EPROMs, mensajes de error, cómo escribir código legible, los errores más comunes que cometen los principiantes, cómo registrarse, explicaciones -paso a paso- para principiantes, bugs conocidos, utilidades suministradas, UNIX, etc.

Las instrucciones completas para instalar la versión de DICE, del disco de portada, están en

la página siguiente. Una vez instalado, podrás oír los programas de ejemplo que lo acompañan. Podrás modificarlos y experimentar con ellos cambiando líneas del código fuente "para ver que ocurre". Te aconsejo que lo hagas, ya que es una muy buena forma de aprender divirtiéndote.

Por desgracia, no podrás compilar programas que usen llamadas a la librería gráfica o a cualquier otra del AmigaDOS, ya que, a la versión de DICE que te hemos suministrado, no le hemos podido añadir los llamados "includes", pues son propiedad de Commodore.

Puedes obtenerlos directamente de Commodore:

Envía un cheque por valor de 25 libras, pagable a Commodore Business Machines (UK) Ltd, a Sharon McGuffie, Commodore Business Machines (UK), Commodore House, The Switchback, Gardener Road, Maidenhead, Berks. SL6 7XA, y pregunta por "The Native Developers Toolkit".

Si estás interesado en obtener los "includes" ponte en contacto con nosotros y haremos los trámites por tí.

Otra forma de conseguir los "includes" de Commodore es comprando la versión registrada de DICE. Puedes encontrar una explicación completa de cómo registrarte en el fichero register.doc que encontrarás en el directorio docs, una vez que descomprimas la versión "Shareware" de DICE.

Por último, existen dos formas más de conseguirlos, aunque son opciones más caras: la primera es comprar el SAS/C y la segunda es comprar el libro en inglés "Complete Amiga C" de Cliff Ramshaw. (Este libro incluye la versión registrada de DICE y además, los "includes" de Commodore).

Hemos preferido llenar el disco de portada al máximo, así que no hemos añadido ningún script para descomprimir DICE; por lo tanto, tendrás que descomprimirlo tú mismo desde una Shell.

DICE está guardado en el disco de portada como dos archivos separados, cada uno de ellos tendrá que ser descomprimido en un disquete aparte. Naturalmente, si dispones de disco duro, podrás instalar DICE en la partición que consideres oportuna.

castellano, sobre cómo programarlas.

❑ No es necesario poseer profundos conocimientos de informática para abordar la lectura de este curso, aunque sí debes estar familiarizado, al menos, con el Amiga y los conceptos más comunes relacionados con él, bien sea de hardware (discos, ficheros, teclado, pantalla, memoria, ...) o de software (conocimientos básicos de AmigaDOS. Concretamente, necesitas conocer los comandos del AmigaDOS: dir, copy, format, delete, type, list, rename, etc. y saber cómo se utilizan los directorios y el acceso a los ficheros).

Encontrarás más información sobre estos conceptos en el manual que acompaña a tu Amiga. También, interesaría que entendieras los conceptos de variable, vector, bucle, subrutina, función, acción, etc.

❑ La mayoría de los ejemplos de programación de este curso no requieren conocimientos matemáticos o técnicos especiales, pero aquellos que sí lo requieren, como por ejemplo programación en 3D, vienen precedidos de un pequeño resumen teórico. Así mismo, se requerirá cierto conocimiento del sistema operativo del Amiga -desde el punto de vista del programador- y del lenguaje ensamblador del 680X0, pero para eso se publican, o se publicarán, también en AMIGA.InFo sus respectivos cursos; de forma que progresivamente te conviertas en un auténtico experto de la programación del AMIGA.

Listados incluidos en el disco de portada

TODOS los ejemplos que aparecen en el curso han sido compilados y ejecutados sin errores. El texto fuente de los mismos, junto

con los ficheros de entrada utilizados y los resultados de salida producidos, están disponibles en el disco de portada.

La gran cantidad de comentarios explicativos que se incorporan en todos los ejemplos, salvo en los más sencillos de este curso, están para tu provecho, con la esperanza de que puedas aprender gracias al ejemplo.

Orígenes del C

El C fue diseñado e implementado por Dennis Ritchie, en los Laboratorios Bell de AT&T, para ser usado originalmente en la familia DEC PDP-11 de Digital Equipment (hacia el año 1972). El desarrollo del lenguaje ha estado fuertemente unido al del sistema operativo UNIX ya que el autor, Dennis Ritchie, simultaneaba el desarrollo de los dos (UNIX junto con Ken Thompson).

El sistema operativo UNIX, en un principio, estaba escrito en lenguaje ensamblador para el mini-ordenador PDP-7 de DEC, pero las constantes necesidades de mantenimiento, adaptación y crecimiento del sistema, llevaron a Ken Thompson a desarrollar un lenguaje independiente de la máquina.

En 1970 Ken Thompson, creó el lenguaje B inspirándose en el lenguaje BCPL desarrollado por Martin Richard en 1967 anticipándose a la moda de los lenguajes estructurados. Ken Thompson desarrolló un compilador y reescribió el sistema operativo UNIX sobre un ordenador PDP-11.

Posteriormente, en 1972 y basándose en el lenguaje B que Ken Thompson había creado a partir del BCPL, Dennis Ritchie definió el lenguaje C. Los lenguajes BCPL y B ya disponían de instrucciones de control clásicas

como: if, while, do, etc. y de conceptos de programación como punteros, asignación dinámica, recursividad, etc. Al lenguaje C se le añadió la noción de tipos de datos, pero de forma menos firme que en los lenguajes ALGOL o PASCAL (anteriores al C).

Ya en 1973 nacía un nuevo UNIX escrito por primera vez en un 90% bajo lenguaje C y diseñado para un ordenador PDP-11. Así, paralelo al sistema UNIX, es como nació el propio lenguaje C, un lenguaje escrito por programadores y con todas las utilidades que un programador necesita de un sistema operativo, a la vez que quedaba totalmente abierto a todas las nuevas necesidades que cada programador precisara.

Durante muchos años, el C estándar fue el proporcionado con el sistema operativo UNIX 5 y descrito en "The C Programming Language" por Brian Kernighan y Dennis Ritchie. Se le conoce a menudo como el "estándar K&R". Tanto el UNIX como el C se han implementado posteriormente en un número cada vez mayor de ordenadores, desde los microordenadores de mayor potencia hasta supercomputadores como el Cray-2, y se están transformando en un estándar industrial para el desarrollo de software. En lo que casi podría calificarse de milagro, la mayoría de estas implementaciones eran altamente compatibles entre sí, a nivel de código fuente.

Sin embargo, debido a que no existía un estándar, había discrepancias. Para cambiar esta situación, a principios de 1983 se creó un comité para trabajar en la creación de un estándar ANSI, que definiría de una vez por todas el lenguaje C. Tras siete años de "trabajo", el estándar fue adoptado en diciembre de 1989.

Gracias a este estándar, se ha

independizado al C del sistema operativo UNIX, soportando en la actualidad la mayoría de sistemas operativos: AmigaDOS, MSDOS, VMS/VAX, SYSTEM 7, OS/2, MACH, etc.

Este curso refleja el estándar ANSI C. Sin embargo, también se cubren importantes aspectos del viejo estándar K&R.

¿Para qué se utiliza el C?

Por lo que hemos visto sobre la historia del lenguaje C, podríamos pensar que es un lenguaje para el desarrollo de sistemas operativos, esto no es exactamente así. Un lenguaje de propósito general, como el C, puede ser utilizado para resolver una gran variedad de problemas informáticos en áreas como:

- ✓Desarrollo de programas ofimáticos (procesadores de texto, bases de datos, hojas de cálculo, etc.).
- ✓Programación de sistemas operativos (UNIX, MACH, LINUX, AmigaDOS, etc.).
- ✓Aplicaciones científicas y de ingeniería (Simulación, Redes, etc.).
- ✓Aplicaciones financieras, Gestión comercial o familiar, etc.).
- ✓Programación de microprocesadores.
- ✓Programas de comunicaciones.
- ✓Aplicaciones de inteligencia artificial o robótica.
- ✓Diseño de juegos y pasatiempos en general.
- ✓Programas de educación.
- ✓Para controlar maquinaria compleja.
- ... y para mil cosas más.

Después de haber leído el párrafo anterior, pensaremos que el lenguaje C sirve para todo tipo de aplicaciones, pero tampoco es del todo cierto. Para resolver determinados problemas disponemos de lenguajes que nos facilitarán el trabajo, por ejemplo, si tenemos que realizar el

mantenimiento de un fichero (altas, bajas, consultas, etc.) la mejor solución es manejar una base de datos desde AREXX. Esta solución, nos ahorrará tiempo ya que, en C, deberíamos de realizar el programa completo. En la base de datos, tendríamos parte del trabajo resuelto. Además, es muy posible que esa base de datos haya sido programada en C, como es el caso de SuperBase 4.

Ventajas del lenguaje C

Cuando se nos planteó el reto de escribir un curso de programación para Amiga, la primera decisión a tomar fué qué lenguaje de programación escoger para los ejemplos. Había varias opciones claras: Modula-2, C, Ensamblador... pero nos quedamos con el C debido a que en los últimos años ha habido una mayor tendencia al uso del C entre los programadores profesionales. Vamos a ver algunas de las razones que hacen del C uno de los lenguajes más potentes y populares:

○ C es un lenguaje de programación estructurado y de propósito general. ¿Cuáles son las características de un lenguaje de programación estructurado?

Pues bien, las dos características principales que distinguen un lenguaje estructurado son, la posibilidad de utilizar subunidades o bloques y la inclusión en el lenguaje de bucles controlados por test de tipo lógico o booleano. Un bloque es un conjunto de sentencias, también llamadas instrucciones, indivisibles que están lógicamente relacionadas, es decir, si una instrucción de un bloque es ejecutada, también deben ser ejecutadas todas las restantes del bloque. Por ejemplo, supongamos una sentencia "if" en la que, si la condición es verdadera, deben ejecutarse tres sentencias simples. Si dichas

sentencias pueden agruparse conjuntamente y referenciarse fácilmente y además se les puede considerar como una unidad indivisible, entonces forman un bloque. En el Ejemplo 1 observará un fragmento de un programa en C que mostrará el número 007 en la pantalla si X es menor que 20.

No existe ninguna posibilidad de que sólo se escriba en la pantalla 0 ó 7 ó 07 o cualquier número distinto de 007, puesto que las instrucciones entre llaves forman un bloque, y si X es menor que 20, se ejecutarán todas; en caso contrario (X>20 ó X=20) ninguna instrucción del bloque será ejecutada.

Ejemplo 1

```
if X < 20
{
  printf("0");
  printf("0");
  printf("7");
}
```

C es un lenguaje estructurado ya que permite la utilización de bloques y la inclusión de bucles controlados por test de tipo lógico o booleano.

A continuación presentamos algunos ejemplos de lenguajes estructurados y no estructurados.

No estructurados	Estructurados
FORTRAN	Modula-2
BASIC	Pascal
COBOL	Ada
ENSAMBLADOR	C

Tabla 1: Pequeña clasificación de lenguajes según si son estructurados o no.

Instalación:

El texto que sigue a continuación es una explicación detallada de cómo debes instalar la versión de DICE, incluida en el disco de portada del primer ejemplar de AMIGA.InFo. Por favor, lee las instrucciones cuidadosamente, antes de intentar instalarlo.

Necesitarás abrir una Shell y usar unos simples comandos (ed, assign, rename y copy). Si no estás seguro de saber abrir una ventana del Shell, consulta el manual suministrado al comprar tu Amiga.

La versión Shareware de DICE, contenida en el disco de portada, funciona con versiones 1.3 y superiores del Sistema Operativo del Amiga. Si todavía estás usando la 1.2, ¿no crees que ya va siendo hora de que actualices tu equipo?

Las instrucciones que debes seguir varían dependiendo de qué versión del sistema operativo estés utilizando actualmente y también de dónde quieras instalar DICE: en disco duro o en disquetes. Sigue el proceso que necesites de los detallados a continuación:

Instalando DICE en 2 disquetes:

Antes de nada, necesitarás dos discos formateados y totalmente vacíos. Puedes formatear los dos discos usando el Workbench, por ejemplo. Consulta el manual de tu Amiga si no estás muy seguro de cómo formatear discos. Una vez formateados, renómbralos como **DiceDisk1** y **DiceDisk2**. Es importante que te asegures de que el nombre de los discos es correcto, o el resto de la instalación no funcionará.

En el directorio llamado DICE, del disco 1 de portada, encontrarás dos ficheros:

dicepart1.lha
dicepart2.lha

Éstos son ficheros comprimidos. Aparentemente, sólo son dos ficheros, pero cada uno de ellos contiene muchos otros. El primer fichero: **dicepart1.lha** contiene todos los ficheros que han de ir en el primer disco que has formateado, **DiceDisk1**. El segundo contiene los ficheros para el segundo de los discos, **DiceDisk2**. Abre una Shell y teclea el comando siguiente. Al final de cada línea presiona la tecla <Return>.

AMIGAINFO1:c/lha x AMIGAINFO1:DICE/dicepart1.lha DiceDisk1:

Sigue las instrucciones que te aparecen en la pantalla, insertando el disco que te pida en cada momento. Si tienes dos unidades de disco, podrás usarlas para acelerar el proceso. Cuando la operación esté completa, y aparezca de nuevo el prompt de la Shell, necesitarás realizar un proceso similar para el segundo disco:

AMIGAINFO1:c/lha x AMIGAINFO1:DICE/dicepart2.lha DiceDisk2:

Una vez acabada la operación, ya dispondrás de dos discos con la versión Shareware de DICE instalada en ellos. No podrás arrancar con ninguno de los dos discos, así que tendrás que arrancar de un disco que contenga el Workbench cada vez que quieras utilizar DICE. Lo más efectivo es que te hagas una copia del Workbench exclusiva para

trabajar con él. La puedes llamar, por ejemplo: **DiceBench**. Puedes realizar la copia usando el comando **diskcopy**, desde la Shell, o usando el Workbench. El comando **relabel** te permitirá cambiar el nombre al disco. Llámalo DiceBench. Éste será tu disco de arranque de DICE.

Si estás utilizando Workbench 1.3:

Arranca desde el nuevo DiceBench. Abre una Shell y teclea:

ed s:startup-sequence

Este comando cargará el editor. Para más instrucciones de cómo manejar este editor, consulta el manual suministrado con tu Amiga. Posiciónate al final del fichero que has cargado (tu startup-sequence), y antes del **endcli** del final del texto, añade las cuatro líneas siguientes:

```
assign dice: DiceDisk1:
assign dlib: DiceDisk2:dlib
assign dinclude: DiceDisk1:include
setenv DCCOPTS "-1.3"
```

Ahora continúa en el punto **Pasos Finales**.

Si estás utilizando Workbench 2.0 o superior:

Arranca desde el nuevo DiceBench. Abre una Shell y teclea:

ed s:user-startup

Este comando cargará el editor. Para más instrucciones de cómo manejar este editor, consulta el manual suministrado con tu Amiga. Posiciónate al final del fichero que has cargado (tu user-startup), y añade las cuatro líneas siguientes:

```
assign dice: DiceDisk1:
assign dlib: DiceDisk2:dlib
assign dinclude: DiceDisk1:include
setenv DCCOPTS "-2.0"
```

Pasos Finales

Graba el texto y sal del editor. Arranca, de nuevo, de DiceBench y, usando una Shell, teclea:

ed s:shell-startup

Y añade esta línea al final del texto:

path dice:bin add

Graba el texto y sal del editor. Por último, teclea esta línea desde una Shell:

**copy DiceDisk1:s/.edrc.floppy
DiceBench:s/.edrc**

Ahora, ya dispones de DICE instalado totalmente en dos discos. Si tienes múltiples unidades de disco, te ahorrarás un gran número de cambios de disco. Cuando arranques del nuevo DiceBench, te pedirá que insertes discos en varios puntos durante el proceso de arranque. No te preocupes, esto es normal. Puedes continuar en el punto **Utilizando DICE**.

○ No es un lenguaje de juguete. El C es un lenguaje de programación suficientemente rico y robusto como para ser una seria herramienta en el desarrollo de software profesional.

○ Está altamente estandarizado debido a que, por fin, tiene una simple y reconocida definición del lenguaje (esta definición estándar se llama ANSI-C).

○ El C es especialmente bueno para escribir programas y aplicaciones de sistema, debido a su facilidad para controlar el hardware a muy bajo nivel. Está considerado como un lenguaje de alto y bajo nivel (para abreviar se dice que el C es un lenguaje de nivel medio). Esto no significa que sea menos potente o más difícil de usar que lenguajes de alto nivel como BASIC o Modula-2, ni implica que C sea similar al lenguaje ensamblador; simplemente combina elementos de lenguajes de alto nivel con la potencia del lenguaje ensamblador. La Tabla 2 muestra cómo encaja el lenguaje C en el conjunto de lenguajes.

Nivel Alto:	Lenguaje Natural Prolog Ada Pascal COBOL FORTRAN BASIC Modula-2
Nivel Medio:	C FORTH Ensamblador
Nivel Bajo:	Lenguaje Máquina

Tabla 2: Comparación entre los lenguajes de programación más populares. El C es un lenguaje de nivel medio.

Por un lado, está considerado como un lenguaje de bajo nivel debido a que proporciona mecanismos que pueden ser utilizados para acceder a los niveles más bajos del ordenador: acceso a los registros, tratamiento de un simple bit, de un byte o de una palabra, manipulación de direcciones y punteros, etc (elementos básicos con los que funciona un ordenador). Por otro lado, está considerado como un lenguaje de alto nivel debido a que proporciona facilidades propias de los lenguajes de alto nivel, como pueden ser: soportar el concepto de tipo de datos (enteros, caracteres y reales. Un tipo de datos define un conjunto de valores que puede tener una variable junto con un conjunto de operaciones que se pueden realizar sobre esa variable), permitir utilizar estructuras de datos muy complejas, permitir casi todas las conversiones de tipos posibles, etc.

PRECAUCIÓN: C no lleva a cabo comprobación de errores en tiempo de ejecución como puede ser la comprobación de límites de vectores, que realizan otros lenguajes de alto nivel. Estas comprobaciones quedan enteramente como responsabilidad del programador.

○ Estos dos aspectos del lenguaje son complementarios. Las facilidades de bajo nivel incrementan la utilidad del lenguaje en cada máquina específica, mientras que, las facilidades de alto nivel permiten independizar el programa del hardware, haciendo posible que el programa se pueda utilizar en diferentes máquinas. El mecanismo de los punteros de C es un buen ejemplo de esta independencia del hardware. Estos permiten al programador incrementar un vector de elementos sin explicitar en el código información sobre el tamaño físico exacto de cada elemento del

vector. El compilador de C será el encargado de generar el código apropiado para el tipo de dato de cada elemento del vector. Características como ésta, hacen del C un lenguaje mucho más transportable que el resto de lenguajes de alto nivel y, ciertamente, más transportable de lo que uno se podría esperar de un lenguaje que permite un contacto tan "íntimo" con el hardware.

Esta característica es una de las más importantes del C y es la culpable del gran éxito que está teniendo. La portabilidad permite que un programa diseñado para ejecutarse en una máquina pueda compilarse y ejecutarse en otra completamente diferente, siempre y cuando se respeten unas pocas reglas durante su programación. Esto presenta la importante ventaja de no tener que ceñirnos a un único fabricante ni modelo, es decir, un programa desarrollado y probado en un ordenador personal bajo sistema operativo AmigaDOS, podría ser compilado y ejecutado sin modificaciones en un ordenador grande, bien bajo sistema operativo Unix u otro sistema cualquiera.

Otra posible combinación puede ser la de coger un fuente en C escrito para Amiga y transferirlo a PC con el "DOSToDOS" o mediante "CROSSDOS", una vez que tengamos el programa en el PC podremos hacer ciertos cambios específicos de los PC's (por ejemplo, los referentes a las diferentes tarjetas gráficas, si es que no hemos respetado la regla de no utilizar características específicas del hardware de la máquina, como son: viewports, copperlists, etc. de Amiga), y compilar el fuente en el PC sin problemas. (es decir, no te hace falta saber nada acerca de segmentos de datos o pila, de cómo se representan los datos en un PC, o cosas por el estilo). El C es, pues, muy superior al BASIC, Pascal o FORTRAN en cuanto a portabilidad.

```
assign dice: Work:Dice
assign dlib: Work:Dice/dlib
assign dinclude: Work:Dice/include
setenv DCCOPTS "-1.3"
```

Ahora continúa en el punto **Pasos Finales**.

Si estás utilizando Workbench 2.0 o superior:

Abre una Shell y teclea:

```
ed s:user-startup
```

Este comando cargará el editor. Para más instrucciones de cómo manejar este editor, consulta el manual suministrado con tu Amiga. Posiciónate al final del fichero que has cargado (tu user-startup), y añade las cuatro líneas siguientes:

```
assign dice: Work:Dice
assign dlib: Work:Dice/dlib
assign dinclude: Work:Dice/include
setenv DCCOPTS "-2.0"
```

Pasos Finales:

Graba el texto y sal del editor. Usando una Shell, teclea:

○ Otro aspecto importante del lenguaje C es que sólo tiene 32 palabras clave (27 del estándar de Ritchie y Kernigham y 5 añadidas por el comité de estandarización del ANSI), que constituyen las órdenes que conforman el lenguaje. Debido a este pequeño juego de instrucciones y sus funciones, los programas desarrollados en C suelen ser de dimensiones bastante reducidas en comparación con otros lenguajes de programación. Por ejemplo, el AmigaBASIC de Microsoft tiene 198 palabras reservadas.

○ Lo compacto del código fuente del C ahorra espacio en disco, reduce el tiempo de compilación y también la cantidad de escritura que ha de realizar el programador.

Por ejemplo, las palabras clave de Pascal o Modula-2: BEGIN y END son simplemente en C los símbolos "{" "}"; la expresión que en Pascal o Modula-2 sería I:=I+1, en C sería I++; y el código en C equivalente al código en Pascal o Modula-2 del Ejemplo 2 sería el mostrado en el Ejemplo 3. En él se ve que el código puede ser escrito en C, con un ahorro del 60% de caracteres.

Una vez que te acostumbres a las sentencias del C, probablemente no desearás volver a la sintaxis verbal de otros lenguajes de programación bastante más extensa.

```
A=A+5;
IF A>20 THEN
BEGIN
  A=A-1;
  ... (* aquí irá el resto del código *)
END
```

Código escrito en Pascal o Modula-2. Los programas escritos en Modula 2 son más legibles que los escritos en C, pero son de mayor tamaño.

```
ed s:shell-startup
```

Y añade esta línea al final del texto:

```
path dice:bin add
```

Graba el texto y sal del editor. Por último, teclea esta línea desde una Shell:

```
copy Dice:s/.edrc.hd s:.edrc
```

Ahora realiza un reset para asegurarte de que los cambios efectuados toman efecto, y bueno, DICE ya está instalado en tu sistema.

Utilizando DICE:

Ya disponemos del compilador totalmente instalado y preparado para trabajar. Si lo has instalado en el disco duro, entonces no tendrás problemas. Si lo has instalado en disquetes, no olvides que tendrás que arrancar desde una copia del Workbench llamada **DiceBench**. El AmigaDOS necesita que introduzcas DiceDisk1 cada vez que utilices alguno de los comandos de DICE: dme, dcc, etc. de los guardados en el directorio **bin**. Si no lo haces así, te mostrará el mensaje de error "command not found". Esto último, sólo es aplicable a los sistemas basados en disquetes.

```
if ((a += 5) > 20) {
  a--;
  /* aquí irá el resto del código */
}
```

Código en C equivalente al del Ejemplo 2 en Pascal o Modula-2. Podemos observar el sustancial ahorro de caracteres que se ha producido.

COMENTARIO: El C se caracteriza principalmente por la escasez de reglas pero, esto tiene una gran desventaja, al poder caer en la tentación de desarrollar programas difíciles de entender y mantener. Ésta es la causa de su mala reputación entre los programadores de Pascal o Modula-2. Se sabe que muchos programadores se precian de explotar la capacidad de compactar las expresiones del C a costa de la legibilidad del programa (Dos muestras de ello son: el libro de C "The Puzzle", y el concurso anual de código rebuscado). ¡No es necesario escribir código incomprensible!

○ C es un lenguaje modular. Los programas pueden escribirse en módulos; cada módulo queda en un fichero aparte y cada fichero se compila separadamente. Luego, cada una de estas unidades, compiladas por separado, pueden ser combinadas (enlazadas) en un programa entero.

Las rutinas utilizadas normalmente pueden colocarse en una librería. Dicha librería podrá ser enlazada en los nuevos programas, cuando sea necesario.

Esta modularidad reduce significativamente los errores de programación e incrementa el desarrollo eficiente del software.

Hay cuatro pasos que tendremos que seguir para escribir un programa en C: editar, compilar, montar (en inglés, linking) y por último, ejecutar.

1. Editar

Todas las operaciones que podemos hacer con DICE, hay que hacerlas desde una Shell. Por lo tanto, lo primero es abrir una Shell.

Lo siguiente es decidir donde vamos a guardar/grabar nuestro trabajo. La RAM es el "dispositivo" más veloz pero, por supuesto, si se produce un guru o apagas la máquina, perderás todo el trabajo realizado. Debido a esto, tendrás que tener un disco formateado y vacío para grabar nuestro trabajo, o un directorio en el disco duro.

Todos los ejemplos del curso están en la forma conocida como "código fuente". Para introducirlos en el Amiga usaremos un editor de textos. El programa **Ed**, que hemos utilizado anteriormente para modificar el startup-sequence, es uno de ellos. Pero DICE tiene incluido un editor mucho mejor, llamado **DME**. Puedes ejecutar DME desde una Shell, tecleando:

```
dme
```

○ C ofrece una enorme librería de rutinas de apoyo en tiempo de ejecución. La mayoría de las implementaciones del C estándar, ofrecen librerías de funciones para: el control entre ficheros de alto y bajo nivel y discos de entrada/salida, control de la pantalla y teclado, conversión de variables, interfaz con el sistema operativo, control de memoria, funciones matemáticas, gráficos, etc.

○ C es fácil de aprender. Admitámoslo: no es tan fácil de aprender como el Pascal o el BASIC, pero, es ciertamente mucho más sencillo que el lenguaje ensamblador ya que tiene menos órdenes que el Pascal y además mantiene la programación estructurada y la longitud de los nombres de los símbolos (recordemos que el lenguaje ensamblador sólo toma un cierto número de caracteres de las variables y etiquetas y lo restante es, simplemente, ignorado).

○ C es un lenguaje compilado, y como tal, permite la protección del software que desarrollan las compañías profesionales, y por supuesto, también protege el código que puedas programar tú, de posibles "miradas no deseadas" de otra gente.

Los programadores de AmigaBASIC o AMOS, se ven obligados a suministrar el código fuente de su programa para que se pueda ejecutar. Es entonces, cuando se pueden producir modificaciones, copias ilegales, infracción del copyright en poder del autor, etc. El C permite obtener un ejecutable que no guarda ninguna relación con el código fuente que lo generó, de esta forma, se consigue proteger totalmente el código del programa. Si el programa estuviera escrito en AmigaBASIC podríamos modificar, sin problemas, la línea o líneas que se encargan de imprimir el nombre del autor y poner el nuestro.

Normalmente, ejecutaremos DME pasándole como parámetro el nombre del fichero de texto que queremos editar. Supongamos que vamos a grabar nuestro código fuente en un disquete llamado **MiCodigo**. Si tu programa se llama ejemplo.c (por convenio, se pone la extensión .c al final del nombre), entonces teclearemos:

```
dme MiCodigo:ejemplo.c
```

(Los usuarios de la versión de DICE en disquetes, tendrán que introducir **DiceDisk1** en la unidad de disco antes de teclearlo). Este comando abrirá el editor con un fichero vacío. Simplemente teclea un listado y después selecciona **<Save-Quit>**. Ya estaremos en condiciones de pasar al siguiente paso.

2 y 3. Compilar y Montar

Compilar es el proceso de traducir el código fuente escrito en C a código máquina, lenguaje ejecutable por la unidad central de proceso del Amiga. La compilación es la función principal de DICE. Antes de poder ejecutar un programa compilado, hay que montarlo. El proceso de montar, enlaza el programa compilado con otros programas y librerías

Si nos colocamos en el extremo opuesto, las ventajas del C se convierten en desventajas. Y, al igual que todos los lenguajes de programación, el C también tiene inconvenientes:

✓ Como el C es un lenguaje compilado, la modificación de programas simples lleva más tiempo del que se emplearía con un lenguaje interpretado, como por ejemplo el Amiga BASIC.

✓ La modularidad del C incrementa el tiempo de compilación en programas pequeños.

✓ Al ser tan conciso se hace menos legible

que el Pascal.

✓ La tradición de grandes librerías de funciones estándar hace, normalmente, que el C sea más caro que otros lenguajes.

✓ Lo compacto y flexible del C hace que algunas de sus partes resulten difíciles de aprender, sobre todo la referente a punteros.

Pero incluso con estas desventajas, el C es, probablemente, el mejor lenguaje para muchos propósitos y programadores.

En el próximo artículo, se introducirán los conceptos básicos necesarios para poder seguir el curso. Se presenta y comenta en el mismo,

nistrado de DICE, está repleta de ficheros de texto (en inglés) que completan la documentación del compilador. Esta documentación en disco te ayudará a usar el sistema de la mejor forma posible. Los ficheros de ayuda se encuentran en el directorio **doc/**. Para los usuarios de disquetes, es este directorio **DiceDisk2:doc/**. Para los usuarios de disco duro, es este otro directorio **dice:doc/**. Por favor, realiza el esfuerzo de leer la documentación (si sabes el suficiente inglés). Te ayudará a solventar muchos de los problemas con los que te vas a ir encontrando, sobre todo, al principio.

Diferencias entre la versión Shareware de DICE y la Registrada

La versión de DICE incluida en el disco de portada, que acompaña al primer ejemplar de AMIGA.InFo, es distribuida libremente. El entorno del compilador y éste funciona totalmente, pero no contiene algunas de las características extras que tiene la versión registrada de DICE. Registrarse cuesta \$50 (dólares); encontrarás los detalles de cómo registrarte en el fichero llamado **Register.doc** situado en el directorio **doc/** de los discos de DICE. Alternativamente, existe la posibilidad de que nos pidas la versión registrada de DICE a nosotros. La redacción de AMIGA.InFo realizará los trámites por tí, sin coste alguno.

Ventajas de la versión registrada:

◆ Incluye el set completo de "Includes" de Commodore y la **amiga.lib** (esencial para sacarle el máximo partido a los gráficos y sonido del Amiga).

◆ Trae una documentación mucho más extensa, con gran cantidad de programas de ejemplo.

◆ Contiene el código fuente de la librería de C.

◆ Soporta los "bitfields" y las opciones: **_chip** y **_geta4**.

◆ Soporta operaciones matemáticas en punto flotante.

una serie de instrucciones para conseguir compilar los ejemplos del curso con los cuatro compiladores de C disponibles para Amiga: SAS-C V6.51, GCC V2.5.8, DICE C V3.0 y por último Aztec C.

Los conceptos y definiciones formales propias del lenguaje C, se introducirán en el siguiente artículo (tercero), para así, poder dar un poco de culturilla general en éste.

Aunque no se traten en detalle los diversos aspectos de la arquitectura de computadores que todo programador debiera conocer, este artículo presenta un resumen de los conceptos y terminología utilizados en los artículos restantes del curso. ●

Esta última ventaja necesita una pequeña explicación. La versión Shareware de DICE no puede trabajar con variables del tipo "float" (es decir, números con parte fraccionaria). Algunos de los ejemplos de este curso requerirán que el compilador soporte la opción de operaciones en punto flotante para poderlo compilar. Si utilizas la versión Shareware, entonces tendrás que modificar los ejemplos para poderlos compilar.

Por ejemplo, cada vez que te encuentres con una variable del tipo float, en uno de los ejemplos del curso, cambiala al tipo int. Además, donde veas los dos caracteres **%f**, dentro de una llamada **printf** o **scanf**, sustitúyelos por los dos nuevos caracteres **%d**. Si hay asignaciones a estas variables, entonces borra la parte decimal de los valores.

Estos cambios permitirán compilar los ejemplos utilizando variables del tipo entero en lugar de las del tipo float. Recuerda que estos cambios sólo son necesarios si utilizas la versión Shareware de DICE. La versión Registrada, o un compilador comercial, podrán compilar perfectamente los ejemplos sin modificaciones.

Includes de Commodore

Los includes de Commodore son una serie de ficheros que, con ayuda de otro fichero llamado **amiga.lib**, permiten al programador de C, sacarle la máxima potencia a las facilidades del Amiga. Con la versión Shareware de DICE, podrás realizar simples entradas y salidas vía una Shell, pero llamar a las librerías de la ROM del Amiga, para abrir pantallas y ventanas, por ejemplo, no te será posible sin los includes de Commodore. Si te vas a tomar en serio la programación del Amiga, necesitarás estos ficheros, y por supuesto, un compilador de C en condiciones.

La forma más segura de conseguir los includes, es poniéndote en contacto con nosotros. Indica claramente tus datos personales y nos pondremos en contacto contigo lo antes posible. ●

AMIGA.InFo (Ref: Includes,DICE)
Apartado de Correos 166
08750 Molins de Rei
Barcelona

AmigaCenter

PLACAS EMULADORAS

Distribuidor Exclusivo para España Emplant Deluxe (SCSI/2 AppleTalks)	75.900
---	--------

PLACAS GRAFICAS Y VIDEO

Retina 24 (1280x1024) Targeta 24 bits	74.900
Retina BLZ3 (Zorro III)	99.900
Encoder Retina (Salidas VC e Y/C)	29.900

V-Lab Motion, Captura y lanzado de imagenes en tiempo real, soporta Video Compuesto e Y/C, efecto croma key, sincroniza sonido

V-Lab paralelo (Todos los Amiga)	58.900
Opalvision (Targeta grafica 24 bits)	69.900

MEMORIA

Simm 4 Mb A4000	29.900
Simm 1 Mb A4000	8.000
Zips 2 Mb Memory Master/RamBox	15.000

AMPLIACIONES A1200/500

RamBox 2Mb hasta 8 Mb externa	29.900
Memory Master A1200 1 Mb+FPU28+Reloj	26.900
Memory Master A1200 5 Mb+FPU33+Reloj	56.900
(Ampliable hasta 9 MB)	

CABLES Y CONECTORES

Cable adaptador disco duro 3,5 a 2,5	4.300
Cable R.G.B. Euroconector A500/2000	3.200
Cable R.G.B. Euroconector A1200/4000	3.200
Cable Serie Modem	2.300
Adaptador Commodore 23 a 15	3.900
Cable 50 Pins SCSI (Interno)	1.900
Cable 50 Pins SCSI (Externo)	2.400
Cable puerto paralelo (Impresora)	1.200

CONTROLADORAS

Fastlane Z3 (Fast SCSI-II) hasta 64Mb Ram	73.900
A4091 (Fast SCSI-II)	56.900
Oktagon 2008 (SCSI-II) hasta 8Mb Ram	26.900
AT-Bus508 (A500 BUS IDE) Hasta 8 Mb Ram	24.900

ACELERADORAS

Warp Engine	
(68040/40MHZ/Fast SCSI-II/256 Mb Ram)	289.000
Commodore 68040/25 Mhz	140.000

C/Argullós, 127
08016 Barcelona

Telefono: 93-276-3806
Fax: 93-276-3080
BBS: 93-350-6195

CD-ROM

CD_Rom NEC 2XI SCSI-II Interno	37.900
--------------------------------	--------

TITULOS CD

Corel Photo CD todos a	4.900
Aminet III Share	3.900
Giga PD III (3 CD's)	8.900
CDPD III	5.590
CDPD IV	5.590
Meating Pearls	3.950
17 Bit Continuation	4.900
Amiga Tools	5.590
Fred Fish Collection (1-1000 / 2 CD's)	5.900
Video Creator (CD-32)	9.900
Gran variedad de titulos CD-32 disponibles	

MODEMS

USRobotics Sportster V.32bis + Fax	37.900
USRobotics Dual Standard 16.8K	69.900
USRobotics Dual Standard 16.8k + Fax	72.900
USRobotics V.34 28.8K + Fax	93.900
Todos los modems incluyen cable y soft.	

SERVICIOS

SERVICIO DE ESCANEADO

(ENTREGA EN FORMATOS PC, AMIGA, MAC)

CAPTURA DE VIDEO

(Entrega en formatos PC, Amiga, Mac)

IMPRESION EN SUBLIMACION DIN A-4

(Se acepta cualquier formato, BMP,GIF,TARGA,TIFF,JPEG,IFF,PCX,etc...)

GRABACION PROFESIONAL VIDEO

(Grabacion a BETACAM de cualquiera de los formatos anteriormente descritos, se aceptan tambien formatos ANIM (AMIGA) Y FLC, FLI (PC))

CURSOS Y CLASES A PROFESIONALES

Cursos individuales a profesionales sobre diversos programas IMAGINE, REAL-3D, LIGHTWAVE, PRO. PAGE, VISTA PRO, MORPH PLUS, WORKBENCH, FINAL WRITER

GRABACION DE CD

Grabacion de disco duro a CD	10.000
Grabacion de CD a CD	6.000

DISPONIBLE PROXIMAMENTE:

VIPER 030,TOCATA 16,CD-1200,MODULO 486 (EMPLANT)

AMIGA 1200 TOWER

!!! A PRECIOS INCREIBLES !!!

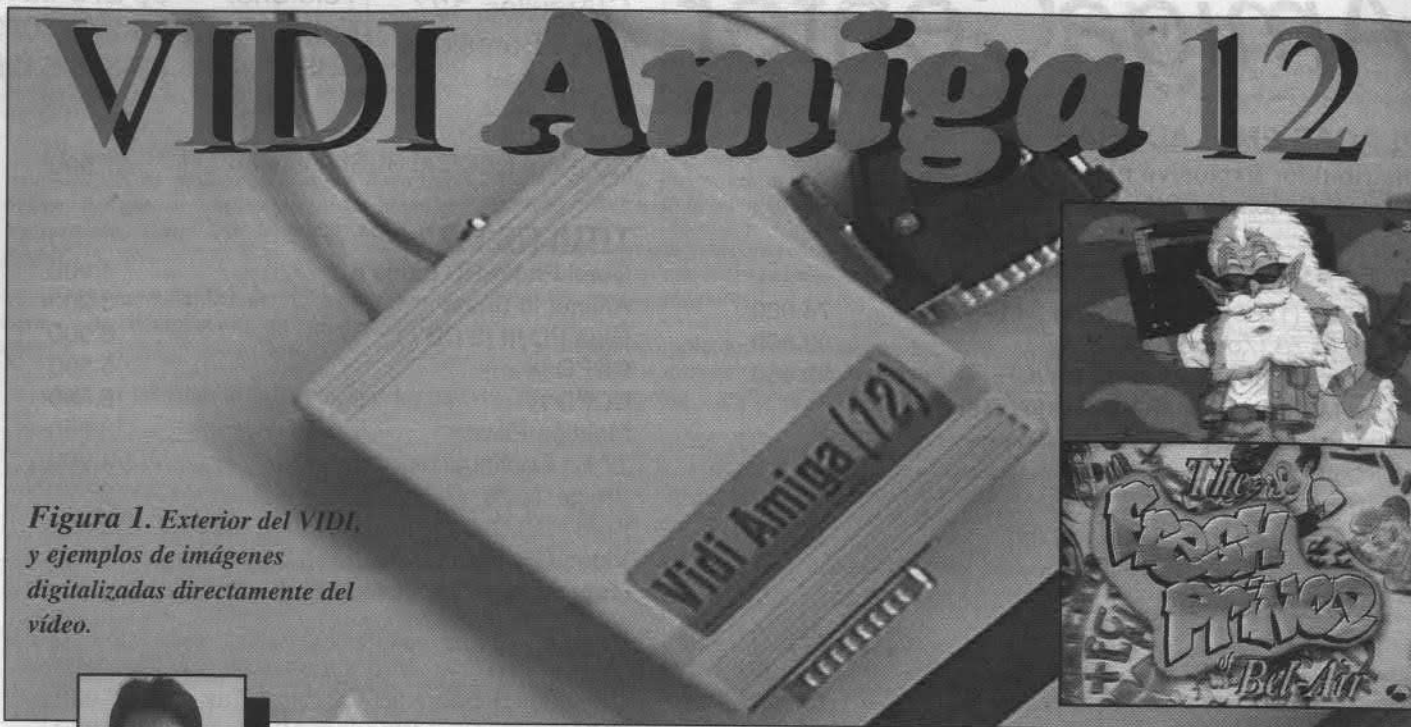


Figura 1. Exterior del VIDI, y ejemplos de imágenes digitalizadas directamente del video.



Atrás quedaron los tiempos en que para digitalizar una imagen en color se necesitaban filtros RGB, una fuente de vídeo de calidad, unos minutos de paciencia y horas de trabajo para grabar animaciones... Alberto Orante nos prueba este mes un económico y rápido digitalizador que consigue resultados más que aceptables ¡en segundos!

El Vidi-AMIGA 12 es un digitalizador de vídeo monocromo de 4 bits, que contiene un separador de colores interno con el que se consiguen imágenes de 12 bits de color, mediante un proceso por software. Éste digitaliza a baja resolución entrelazada o no entrelazada, trabajando, en este último caso, a tiempo real, pudiendo digitalizar secuencias de imágenes y grabarlas como una animación.

Anteriormente, el fabricante ROMBO sacó al mercado el digitalizador VIDI y un separador de colores, que se vendían por separado o juntos en un pack. Eran prácticamente lo mismo que el Vidi-AMIGA 12, pero al estar separados, era más incómodo trabajar con ellos, además de que salía más caro.

Actualmente, se pueden encontrar el Vidi-AMIGA 12 RT (Real Time) que digitaliza en tiempo real, imágenes de 12bits de color, y Vidi-AMIGA 24 RT que digitaliza imágenes de 24 bits en tiempo real además de incorporar la mejora de poder trabajar en alta resolución y en overscan.

PRESENTACIÓN

El Vidi-AMIGA 12 viene en una caja demasiado grande para lo que es en sí su contenido y poco práctica, por la dificultad de acceso a su interior. Una vez abierta, encontraremos un sobre con el manual, registro y un disco con software. Debajo, un cable RCA-RCA y el Vidi-AMIGA 12 colocado en un cartón que cumple, a duras penas, la misión de sujetarlo e impedir que se golpee.

Nota: RCA, también llamado PHONO, es un

tipo de conector empleado en la salida de audio y vídeo del Amiga.

INSTALACIÓN

El Vidi-AMIGA 12 es una caja de plástico, de color negro (o beis en su última versión), de reducidas dimensiones y poco pesada.

En la parte frontal vemos el conector de 25 pins, que va al puerto paralelo. Detrás del digitalizador tenemos un conector RCA hembra para la entrada de la señal de vídeo compuesto, ya sea color o monocroma, y un conector S-Video para conectar una fuente de Super VHS o Hi8. A un lado del Vidi-AMIGA 12 existe un orificio, gracias al cual podemos modificar el brillo. Mediante un pequeño destornillador podemos girar una resistencia variable que se encuentra en él. Tendremos que usarlo cuando el rango de brillo que necesitamos modificar sea demasiado grande como para poder hacerlo mediante el soft.

En el interior del VIDI 12 encontramos dos placas colocadas una encima de la otra, unidas por un conector de 8 pins. La placa inferior se encarga de separar los colores y mandar la imagen a la placa superior (Figura 2) que se encarga de la conversión Analógico-Digital, destacando ésta por su gran número de chips.

En los ordenadores A500, A600 y A1200 se instala conectándolo al PUERTO paralelo directamente como un digitalizador de audio,

y al port de la unidad de discos externa a través de un cable con su conector correspondiente, de donde extrae el voltaje que necesita para funcionar (ver Figura 1).

Debido a su anchura, la caja se solapa con el puerto serie, por lo que tendremos que optar por extraerlo cada vez que tengamos que utilizar el puerto serie, o comprar un cable extensor del paralelo para conectar el Vidi. Esto mismo nos pasará con los Amigas que tienen el teclado separado del ordenador (2000,3000,etc.), ya que éstos tienen el conector de la unidad de discos al lado del conector paralelo. Tendremos obligatoriamente que comprar un cable extensor del paralelo para poder usarlo en estos modelos de Amiga.

Como siempre, y como recuerda el manual, a la hora de instalar cualquier conector, el Amiga deberá estar apagado, ya

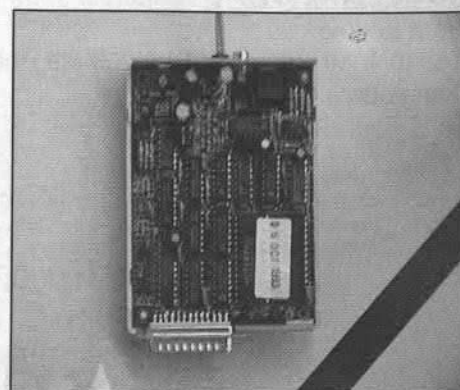


FIGURA 2. Interior del digitalizador VIDI

que si se hace lo contrario, se puede provocar una seria avería en el VIDI o lo que es más probable y peor, en el Amiga.

Después de conectarlo al ordenador, hay que emplear una fuente de vídeo, que puede ser cualquier equipo que contenga una salida de vídeo compuesto, S-VHS o Hi8, y su correspondiente cable.

El software, podremos ejecutarlo sin problemas, desde el disco, siempre que poseamos un ordenador con el Kickstart 2.0 o superior. Esto no será problema si tenemos una versión inferior ya que en el contenido del disco viene un fichero de instalación para los usuarios de un Amiga con el Kickstart 1.3, que añade unas librerías al disco.

Los usuarios de disco duro que deseen instalar el programa en el mismo, o desde el Workbench que tienen instalado, tendrán que ejecutar el fichero de instalación para su correspondiente versión del Kickstart, que únicamente copia las librerías necesarias para poder hacer funcionar el programa. Luego, tendremos que copiar el fichero del programa con su icono, en el directorio de la partición que queramos.

1,2, 1,2, PROBANDO...

Una vez ejecutado el programa, vemos la ventana de la figura 3, donde podremos modificar y grabar las preferencias, leer una imagen o una animación del disco, grabar la imagen o animación con la que se esté trabajando, o meterse en los menús de digitalización (GRAB), transformación (MIX), edición (EDIT), y buffer de imágenes (CAROUSEL). Estos menús aparecen en la parte inferior de la pantalla, visualizando en el resto, la imagen o imágenes con las que estemos trabajando. Se puede, de igual forma, acceder a los otros menús directamente sin tener que regresar al menú principal.

En GRAB observamos la imagen que entra en el digitalizador en monocromo en la pantalla, opción muy útil para conseguir centrarla. Aquí encontramos otras opciones para digitalizar en monocromo, con filtro de rojo, verde y azul, o en color (la suma automática de las imágenes con los tres filtros). También encontramos los controles de brillo, contraste y color, el tiempo de captura entre frame y frame y los modos de digitalización automática, manual y continua, ésta última para digitalizar una secuencia de imágenes.

En MIX podemos adaptar la imagen digitalizada a la resolución que queramos (HiRes/S-HiRes, PAL/NTSC, Lace y Overscan) y con los colores que queramos (16,16 B&W,32,64 EHB y HAM,y modos AGA:64,128,256,256 B&W y HAM8). También encontramos opciones como Optimal Palette y Error Diffusion que en teoría mejoran el resultado pero ralentizan el proceso.

En EDIT podremos hacer 13 efectos variados, que pueden ser como suavizado a 3 niveles, mosaico, efectos a la paleta de colores, resaltadores de contornos y perfiles, etc.

En CAROUSEL (Figura 4), se nos muestra todas las imágenes contenidas en memoria, pudiendo reproducirlas como si fueran una animación, o almacenarlas en un disco como una imagen en formato ILBM, 12 bits, 24 bits, BMP, TIFF, o animación.

Sin duda, es un programa muy completo, pero falla a la hora de transformar la imagen con modos inferiores a 4096 colores debido a que selecciona menos colores de los que soporta el modo de pantalla elegido. Además, realiza una selección muy poco acertada de los colores de la paleta. Existe la opción Optimal Palette con la que no se obtienen resultados algo más satisfactorios. Por este motivo, tendremos que adquirir un programa adicional, como el Art Department Pro, que

solucione este problema cuando necesitemos imágenes en modos inferiores de 4096 colores.

Las imágenes transformadas a HAM8 se ven un poco mejor que con HAM6 aunque en las buenas digitalizaciones no se puede apreciar diferencia. Es recomendable almacenar las imágenes, si no tienen que sufrir ninguna transformación, en HAM6 o si es el caso contrario, en 24bits ya que el formato 12bits es poco estándar y puede dar problemas a la hora de importarlas posteriormente en algunos programas de tratamiento de imágenes.

El Vidi-AMIGA 12 trabaja muy bien con el A1200 ya que los procesos de mezcla de las imágenes RGB, y efectos especiales, van más rápidos que en el A500, aunque el tiempo de digitalización es igual en ambos. Otro motivo es la memoria: el 1200 con 2 MB puede digitalizar hasta 30 imágenes monocromas, o 7 imágenes en color y en un 500 con 1 MB, 10 monocromas o 2 imágenes en color.

Con el Vidi-AMIGA 12 podemos digitalizar una animación a tiempo real con una videocámara en color o monocroma, pero obtendremos unos resultados poco satisfactorios, porque digitalizará dos frames por segundo y observaremos una animación con cambios bruscos de frames. Si queremos obtener un buen resultado tendremos que grabar la secuencia en vídeo y reproducirla a cámara lenta o frame a frame para digitalizarla con las opciones pertinentes. Si queremos realizar una animación en color, tendremos que hacer lo mismo reproduciéndolo frame a frame.

Para digitalizar, desde una cámara de vídeo, una imagen impresa en papel o cualquier otro objeto estático, necesitaremos un trípode para colocar la cámara en posición vertical. Podremos adquirirlo en cualquier establecimiento de fotografía profesional. Si la



FIGURA 4. La opción Carousel es una de las más potentes puesto que te permite visualizar las imágenes que componen una animación como iconos e incluso editar el orden de aparición en pantalla, con tan solo marcarlas con el ratón.

videocámara es monocroma y queremos digitalizar en color tendremos que adquirir unos filtros de colores rojo, verde, y azul. Para obtener buenos resultados, necesitaremos iluminar la imagen con una luz blanca, sin que provoque sombras, brillos o reflejos.

Si lo que vamos a digitalizar viene de un videoreproductor y ha de ser en color, le tendremos que dar al digitalizador una señal óptima si queremos obtener un resultado aceptable por lo que tendremos que conseguir un vídeo de gama alta, preferiblemente con una pausa digital.

Si queremos digitalizar en monocromo, es indiferente la pausa y la cantidad de cabezales que tenga el vídeo, siempre que la imagen tenga un mínimo de calidad.

En las pruebas que realizamos, dependiendo de la imagen, era aconsejable aumentar el valor del contraste, ya que se gana más definición, y el brillo lo situábamos en una posición no muy alta. El color venía bastante saturado, y lo bajábamos una posición.

Tanto las videocámaras como reproductores de vídeo, si son Super VHS o Hi8 y lo conectamos en la entrada de S-Video del Vidi-AMIGA 12, obtendremos mejores resultados ya que este tipo de vídeos dan más líneas y por tanto más definición que los vídeos convencionales, consiguiendo una calidad mayor en la digitalización. Con El Vidi-AMIGA 12 no nos proporcionan un cable para dicha conexión. Tendremos que comprar uno en caso de necesitarlo.

El manual tiene el inconveniente de estar en inglés, pero esto no será un gran problema dada la sencillez del software y la facilidad de instalación. Es un manual breve pero cubre todas las opciones: Contiene todo lo referente a la instalación, un tutorial de los puntos más importantes del funcionamiento del hardware y

el software, una referencia detallada del software, además de otros comentarios de utilidad.

CONCLUSIONES

□ VIDI es una alternativa muy interesante para cualquier usuario que se inicia en el campo del vídeo, puesto que es económico y los resultados obtenidos son aceptables. Sin embargo, para extraerle el máximo partido, es necesario tener un equipo profesional de vídeo.

□ Trabaja sólo en baja resolución, un gran inconveniente, para trabajos en los que se necesite digitalizar imágenes de vídeo monocromas con bastante resolución. Por ejemplo en autoedición, no será útil para obtener una buena calidad en la impresión de las imágenes.

□ El programa de control es completo, sencillo de manejar, pero tiene el gran fallo de paleta, que ya se ha comentado, a la hora de transformar imágenes a modos inferiores de 4096 colores.

□ El cable que viene incluido es muy corto. En mi opinión, tendrían que haber diseñado el Vidi-AMIGA 12 de forma que respetara el conector serie, en el caso del A500 y similares, y el de la unidad de discos externa en el caso del A2000 y similares, o venir con un alargador del paralelo incorporado.

□ Es rápido digitalizando, en comparación con otros digitalizadores de vídeo más antiguos, y es fácil crear una animación si se tiene memoria, para poder ser utilizada en aplicaciones multimedia.

□ Se puede utilizar perfectamente con un Amiga 500/2000 con un mínimo de 1MB de memoria RAM y en todos los nuevos Amiga AGA ●

MATERIAL UTILIZADO

- ◆ Amiga 500 / Amiga 1200
- ◆ Vidi Amiga 12
- ◆ Panasonic Video Cassette Recorder
- ◆ NV-J45 HQ PAL Panasonic Video
- ◆ Cassette Recorder NV-FS90 HQ
- ◆ SVHS PAL HiFi STEREO Sanyo Digital Video Recorder VHR-D6700
- ◆ Sony Triniton HVC - 4060P

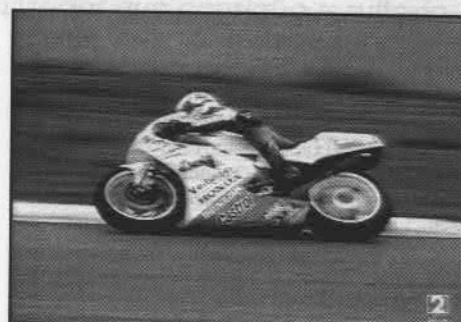
ESPECIFICACIONES TÉCNICAS

- ◆ Nombre: Vidi-AMIGA 12
- ◆ Fabricante: ROMBO Productions Ltd.
- ◆ Resolución de captura:
 - PAL 320x256
 - 320x512
 - NTSC 320x200
 - 320x400

- ◆ Resolución de la paleta:
 - 12 bits (4096 colores)
 - 4 bits Red
 - 4 bits Green
 - 4 bits Blue

- ◆ Tiempo de captura:
 - Monocromo: 1/50 seg. (tiempo real), (no entrelazado).
 - Color: menos de 1 segundo, (no entrelazado).

- ◆ Entradas:
 - Vídeo compuesto, tipo RCA.
 - S-Video (S-VHS, Hi8)



FRONTIER <ELITE II>



El juego viene acompañado de tres magníficos manuales (eso sí, en inglés) y dos discos. La presentación, empaquetado y documentación son dignos de elogiar, e incluso viene acompañado de un mapa espacial de tamaño gigante, con todas las zonas que podemos visitar. Hay horas y horas de juego, y cada partida puede ser totalmente diferente...

FRONTIER - Elite II

Programador/es: David Braben.
Editor/es: GameTek & Konami.
Requiere 1 Mb de memoria.
Instalable en disco duro.
Compatible ROM 1.X / 2.X / 3.X

Abrió un canal de comunicación y pidió permiso para despegar. La nave se elevó rápidamente y encaró el espacio abierto, lista para saltar al hiperespacio. Sabía que en pocos minutos podría encontrar la muerte y, sin embargo, pulsó el botón de salto con firmeza y resolución. Si iba a morir, era preferible hacerlo llevándose con él a tantos enemigos como le resultara posible...

El comandante Aaron Marcus comprobó el inventario de carga: doce toneladas de grano, cinco de robots, tres de licor y nueve toneladas de fuel de hidrógeno. Todo correcto. La lista de embarque volvió a ser revisada: un único pasajero, Andrew Miller. Destino: el Sistema Solar. La carga no suponía ningún rico botón para cualquier pirata espacial, pero Miller complicaba lo que en principio debería ser un viaje tranquilo. Miller era buscado por el Imperio, acusado de espionaje, y la Armada Imperial no se iba a cruzar de brazos dejándolo escapar. Sin embargo, el comandante Marcus aceptó llevarlo. Miller le ofreció una suma de créditos que Marcus no podía rechazar, necesitaba ese dinero con desesperación. Los últimos saltos a los sistemas fronterizos habían dejado a Marcus con poca salud y menos créditos. El comandante realizó los últimos ajustes necesarios y se acomodó en su cabina.

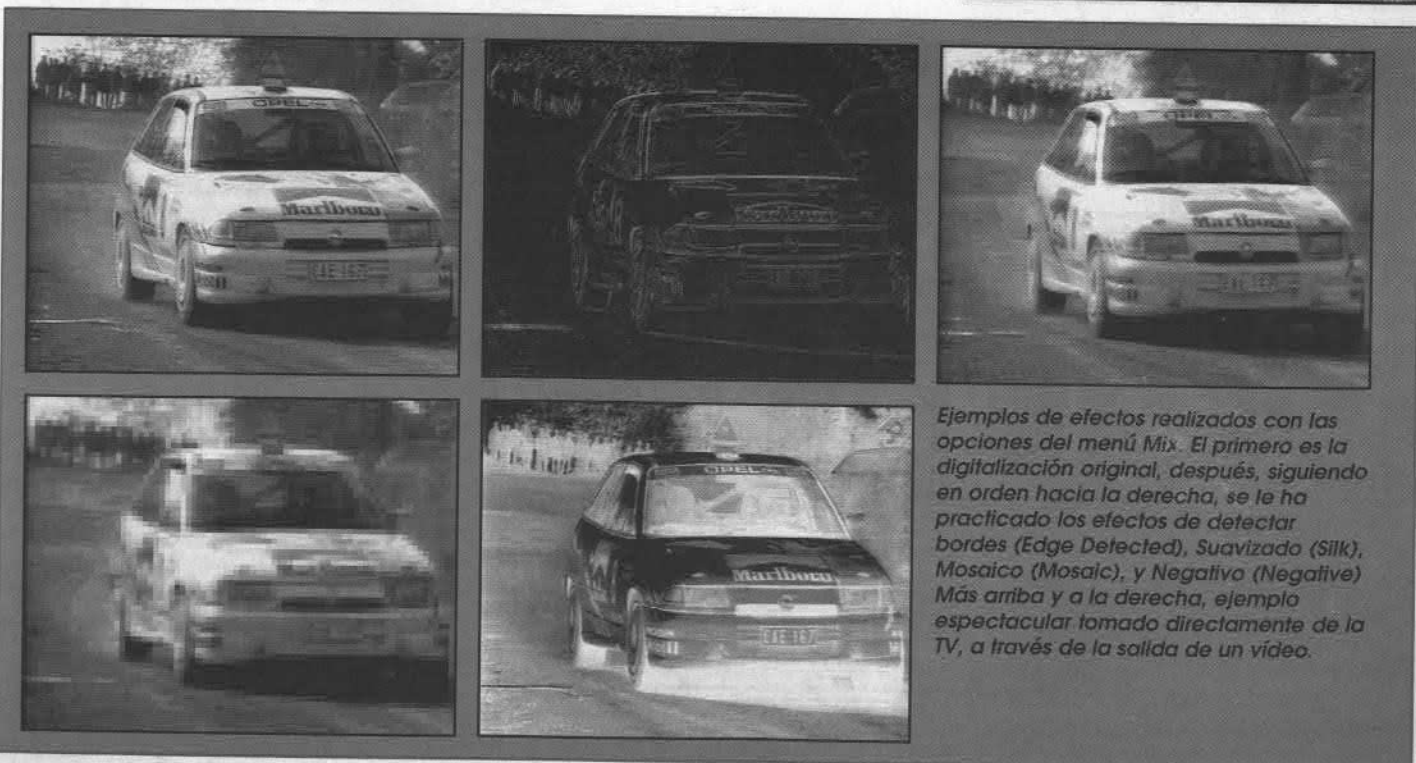
Bien, después de esta breve introducción, pasemos a comentar el juego que ocupa el número uno de nuestro ranking este mes. Frontier es un juego difícilmente clasificable, en él se mezclan el simulador de vuelo y combate con el simulador de "vida", todo ello aderezado con una pizca de estrategia. En Frontier eres el comandante de una pequeña (en principio) nave estelar. Tu objetivo: enriquecerte, adquirir naves mejores y alcanzar el grado de élite tanto en combate como en comercio. Para ello puedes convertirte en un mercader, un contrabandista, un pirata, un asesino, un mercenario, un soldado... Las posibilidades son prácticamente infinitas, así como el espacio dónde desarrollar tus acciones: sistemas de la federación, imperiales, corporativos, independientes... El límite es tu nave. Al principio tus posibilidades son muy limitadas: un pequeño caza de combate y cien créditos es todo lo que posees en este universo, dónde la vida no es nada fácil y la muerte

Cuando un juego te permite multitud de posibilidades diferentes de jugar, es adictivo y aprovecha las capacidades del Amiga, se convierte en algo más... se convierte en Frontier. Joan López nos comenta este mes la segunda parte del juego que convoca reuniones y ligas de miles de aficionados a él...



acecha en el lugar más inesperado. En cada espacio-puerto puedes comprar y vender todo tipo de mercancías: alimentos, metales, productos industriales, incluso existe un mercado negro donde puedes encontrar todo tipo de productos ilegales. Dependiendo del sistema en que te encuentres pueden ser drogas, esclavos, armas, etc. Cada sistema tiene una lista de productos que importa y exporta, así como la lista de productos ilegales. Si decides escoger el camino de la delincuencia deberás enfrentarte a la policía local. Sin embargo, en Frontier, como en la vida real, no siempre los buenos visten de uniforme.

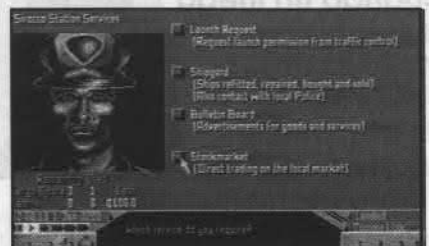
Para comerciar o combatir dispones de una amplia variedad de naves, cada una con unas características diferentes que las hacen más adecuadas para luchar o vender productos en el rincón más remoto de la galaxia. Sin embargo, esto no acaba aquí. Una vez has adquirido una nave puedes equiparla con todo tipo de accesorios: láseres de diferente tipo, motores con mayor alcance y potencia, escudos, cabinas de pasajeros, misiles inteligentes... Esto hace que uno pueda configurar la nave a su gusto y necesidades. No es lo mismo transportar diez toneladas de fruta que tres de armas pesadas. Si la carga es valiosa o ilegal sufrirás mayores ataques por parte de piratas estelares. Si tu misión es asesinar a un ejecutivo corporativo, es aconsejable que tu nave vaya equipada con misiles navales, láseres potentes, escudos y un buen motor. La variedad de acciones y la libertad que posee el jugador para hacer lo que le plazca es lo mejor de Frontier. No estás



Ejemplos de efectos realizados con las opciones del menú Mix. El primero es la digitalización original, después, siguiendo en orden hacia la derecha, se le ha practicado los efectos de detectar bordes (Edge Detected), Suavizado (Silk), Mosaico (Mosaic), y Negativo (Negative). Más arriba y a la derecha, ejemplo espectacular tomado directamente de la TV, a través de la salida de un vídeo.



obligado a cumplir las acciones que te impone el programa, es el programa el que se adapta a ti. Debido a ello, Frontier es el programa de mayor adicción aparecido hasta ahora, no puedes dejarlo y siempre intentarás cumplir aquella misión que parecía imposible, o tratarás de conseguir esa nave de tus sueños con la que serías el rey de los mercaderes.



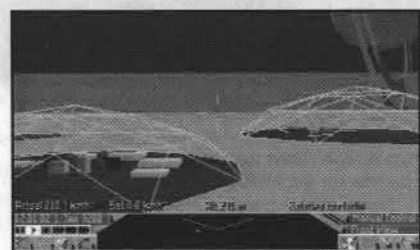
Técnicamente, Frontier es una auténtica maravilla. Las rutinas vectoriales tridimensionales permiten la creación de objetos curvados, con luz y sombras y un nivel de detalle altísimo, siendo éste ajustable por el jugador según sean sus preferencias. Los sistemas planetarios están creados de forma increíblemente real, no en vano David Braben es un apasionado de la astronomía: estrellas rojas, enanas blancas, asteroides, planetas estériles, gigantes gaseosos, etc. Cada sistema es un universo en sí: el número de planetas, satélites, ciudades orbitales, períodos de rotación, astro-puertos, ciudades, distancias entre planetas, todo es diferente en cada sistema planetario. El único punto flaco del aspecto gráfico es la suavidad de movimiento. Mientras que en un A1200 el movimiento es suave, en ordenadores inferiores el movimiento es brusco y difícil de controlar debido a la inercia (otro aspecto más de realidad). Esto afecta principalmente (y únicamente a mi juicio) a las secuencias de combate. Es realmente doloroso ver cómo tu nave es reducida a polvo estelar una y otra vez debido a la poca suavidad con que se mueve. Mas esto no quiere decir que sea imposible y necesites de un A1200 o superior para poder disfrutar de Frontier.

Como en la mayoría de simuladores, el sonido es la parte que menos destaca, pero en Frontier, además de los sonidos propios de la nave, existe una serie de melodías de música

clásica que acompaña tus acciones: el "Danubio Azul" a la hora de atracar, la "Cabalga de las Valkirias" en el momento del combate o al saltar al hiperespacio, etc. Además, el jugador puede apagar totalmente sonido y música si lo desea, o hacer que suene cierta música según la situación, o eliminar ese sonido del motor tan molesto u otros sonidos. Esta flexibilidad suple la falta de calidad que hace que la música sea sólo adecuada y no espectacular.

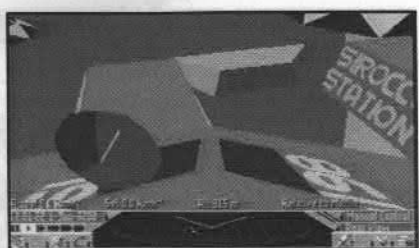


La dificultad es otro punto fuerte de Frontier, aunque nadie te manda meterte allí donde tus



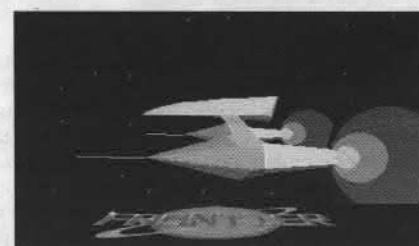
posibilidades de supervivencia son prácticamente nulas. El juego en sí no es difícil de controlar y uno se acostumbra en seguida. Esto no quiere decir que una misión de asesinato sea un paseo o un objetivo imposible. Cada acción tiene un grado de dificultad asociado que el jugador no debe ignorar, así que no tratéis de emular a Han Solo en los inicios de la partida. Con inteligencia y paciencia podréis progresar más rápidamente que yendo a lo loco. Avisados quedáis los jugadores de gatillo fácil, cuesta muy poco crearse poderosos enemigos.

Como había comentado antes, la adicción y la jugabilidad son el plato fuerte de Frontier. Si caéis en sus garras os resultará imposible escapar. Cada vez deseareis ser más poderosos, y ello conlleva horas y horas de largas partidas. Sin



embargo, te verás recompensado cuando tus amigos contemplen atónitos y con envidia la última nave que acabas de adquirir o tu colección de medallas por un servicio fiel a la Federación.

En resumen, éste es un juego que no debe faltar en la colección de todo buen jugador que se precie de serlo. Si hace siete años alucinaste con Elite (recordemos que ésta es la segunda parte), Frontier no te defraudará. Si es la primera vez que lo ves, caerás igualmente ante su poder. Con Frontier la diversión está asegurada durante los próximos meses de tu vida. Recuerda que un juego sólo será bueno si es capaz de entretener durante mucho, mucho tiempo. ¡Y Frontier es capaz de esto y mucho más! ¡Compruébalo tu mismo!. Para los que estéis deseosos de darle al joystick, pídalo a tu distribuidor más cercano, y si no lo tiene, pregunta en MailSoft (Tlf: 93 412 63 10) o en Tu Amiga Ordinadors (Tlf: 93 560 76 12).●



Bienvenidos todos a Top-Demos, la sección de Amiga.InFo donde serán comentadas por Joan López las demos más actuales y cañeras. En este número revisaremos las siguientes: 'Sequential', del grupo Andromeda, y 'Arte', de Sanity. Ajustaos los cinturones y poned el respaldo de vuestro asiento vertical porque allá vamos...

DEMOS

Nombre: Sequential
Grupo: Andromeda
Tipo: Trackmo (1 disco)
Programador/es: HeadX. Código adicional por Dr. Jekyll, Mr. Hyde y Decker
Grafista/s: Archmage, Fairfax y Quest. Gráficos adicionales por View e
Interphase
Músico/s: Mr.Man

Impecable en su diseño como las anteriores producciones de Andromeda (Mindriot, Seven Seas), Sequential es una deliciosa mezcla de programación inteligente y eficiente con unos increíbles gráficos, todo ello acompañado por una música que ayuda a conseguir una de las "trackmos" (demo que carga directamente, sin pasar por sistema operativo) más vistosas de esta temporada. Vayamos, pues, por partes.

Últimamente, todo buen programador que se precie de serlo debe tener un buen conocimiento de matemáticas, y HeadX nos muestra el suyo en una serie de efectos que si bien no parecen rompedores, son más de lo que aparentan a simple vista: vectores de movimientos rápidos y suaves unidos a leyes físicas, como el imán que atrae "vectorballs" y gracias a su velocidad de rotación consigue desprenderse de ellas, el torus (donut) con una textura a cuadros móvil, una pirámide que se refleja en las caras interiores de un cubo o el árbol que crece y se cimbrea debido a la fuerza del viento. Realmente son necesarios conocimientos de física y matemáticas avanzadas para lograr todos estos efectos y los demás que aparecen en la demo, como los objetos vectoriales con "morphing" a la vez que rotan o las letras de gelatina del inicio de la demo. Impresionante.

En el apartado gráfico, realmente no tengo palabras para describirlo. Si habéis visto la demo "Seven Seas" me comprenderéis, y si no, ya estáis consiguiéndola ahora mismo, es todo un placer para la vista. Como ejemplo, echad un vistazo a las imágenes de esta sección...todas son de la demo!. En "Sequential", además del genial Fairfax, grafista de "Seven Seas", intervienen Archmage y Quest, que no tienen nada que envidiar a Fairfax. Logos atractivos e ilustraciones geniales son firma de la casa; parece mentira que con un ratón y un programa de dibujo puedan lograrse estas maravillas. Andromeda ha situado el listón muy alto para todos los grupos que presumen de buenos gráficos.

La música es el aspecto que menos resalta en esta demo, pero no por su mediocridad, sino porque cumple con un papel unificador. Recordemos que esta demo no es un "musicdisk" (o disco de recopilación de módulos de música, en el que el apartado gráfico está relegado a un segundo plano), y que al ser el código y los gráficos los apartados más notorios, la música no impacta tanto como éstos. Sin embargo, sería injusto no reconocer el trabajo de Mr. Man, así que no esperéis que la banda sonora de "Sequential" sea el bombazo del año, pero tampoco la despreciéis por ello.

En resumen: una buena demo que no debe faltar en la colección de todo buen demoadicto que se precie de serlo. Si ya habíais visto alguna demo de Andromeda, ésta no os decepcionará, y si aún no habéis visto ninguna, éste es un buen momento para empezar a conocerlos.



JUEGA CON VENTAJA

¿Cansado de caer siempre en la misma fase de tu arcade favorito? ¿Harto del enemigo de fin de fase que aparece en tus peores pesadillas? ¡No busques más! Esta sección es la solución a tus problemas. Aquí encontrarás trucos y consejos para sacar el máximo partido a tus juegos.

SYNDICATE

Escribe los siguientes códigos en el nombre de la compañía en la pantalla de configuración:
NUK THEM Empezar en el país que quieras.
OWN THEM Todos los países son tuyos.
MARKS TEAM Obtienes el mejor equipo con montones de cyborgs.
MIKES TEAM Equipo normal.
WATCH THE CLOCK Tiempo transcurre rápido.
ROB A BANK Empiezas con 10 millones.

CHAOS ENGINE

Escribe estos códigos para empezar con el máximo de vidas, fuerza, velocidad y montones de dinero:

MUNDO	CODIGO
2	XJKSOMH1Z92V
3	C4HQYOQ4NJ3M
4	V7YRZK8D13GS

EPIC

Pulsa abajo, derecha y ENTER para incrementar al máximo la energía, escudos y armas.

ZOOL 2

Los siguientes códigos deben teclearse en la pantalla del título cuando ZOOL 2 ha sido dibujado y Press Fire for Options está intermitente. Si cuando introduces el código la pantalla no destella, prueba de nuevo.

TRUCOS	CÓDIGO
CREAMOLA 10 vidas.	
VISION 20 vidas.	
TOUGH GUY ... Invulnerabilidad.	
OLDENEMY Tiempo ilimitado.	
ALCENTO 99 objetos recogidos.	
KICKASS Bombas inteligentes ilimitadas.	
BUMBLEBEE ... Salta pantallas (pulsa RETURN).	
NIVEL	CÓDIGO
Swan Lake	SESAME
Blueberry Hill	RONSON
Tooting Common	FUNKYTUT
Snaking Pass	HISSTERIA
Mount Ices	7SLURP
Mental Block	PLUNGER

PROJECT X

Para poder pasar la zona de misiles del primer nivel, antes de entrar en ella ponte en la parte derecha de la pantalla. Permanece ahí y los misiles ni te tocarán. Recuerda volver atrás al acabar la zona.

Nombre: Arte
Grupo: Sanity
Tipo: Trackmo (1 disco)
Programador/es: Mr. Pet. Código adicional por Chaos y McDeal.
Grafista/s: Ra.
Músico/s: Moby.

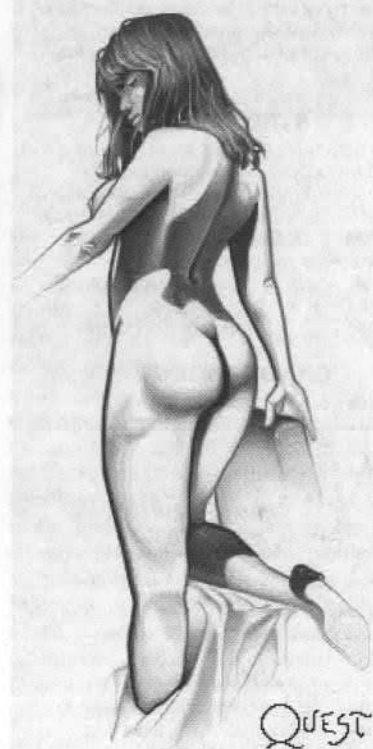
Nunca el nombre de una demo se había ajustado tanto a su contenido. "Arte" es una auténtica pieza de idem, una obra maestra. Impecable en su diseño y su contenido, como es habitual en todas las producciones de Sanity, "Arte" ofrece todo un espectáculo tanto gráfico como musical a quien la contempla.

Aunque mucha gente juzga a una demo por sus gráficos o su música únicamente, esto no sirve de nada si el código no es efectivo u original. Sanity es uno de los actuales grupos punteros tanto por una cosa como por la otra. No olvidemos que Sanity fue el creador (concretamente Chaos) de los "zooms" y deformaciones de gráficos a tiempo real, y todo ello en un Amiga 500. Así queda patente en Arte: el túnel del principio de la demo en el que el suelo y el techo son gráficos que se mueven con una velocidad y una suavidad pasmosa, todo a tiempo real, o las deformaciones de un gráfico rotando y con "zoom" también a tiempo real. Tampoco las rutinas de objetos vectoriales se quedan cortas: el efecto que simula los edificios de una ciudad vista desde arriba con perspectiva, la esfera cortada a tiras y con foco de luz o los objetos de la parte final.

Gráficamente, Sanity es uno de los grupos que más fuerte pegan actualmente, y más ahora con la incorporación de Ra. "Arte" contiene una serie de dibujos surrealistas de gran calidad, desde los personajes más estrafalarios a los logos más vistosos. Mas no acaba aquí la parte gráfica: Ra nos deleita con su habilidad en el dibujo de Albert Einstein o la escena de los dos esqueletos. Y no sólo destaca "Arte" en los gráficos, su diseño es sencillo pero efectivo y divertido a la vez. Pocas demos logran alcanzar el impacto visual que ha logrado "Arte".

En el aspecto sonoro, "Arte" es todo un éxito también. En lugar de recurrir a una melodía "techno" o máquina, como la mayoría de las demos actuales, Sanity apuesta por una melodía menos convencional y más pachanguera, sobre todo el primer módulo, atractivo y adecuado a los gráficos surrealistas. No, la música de "Arte" no es 'rara', es diferente. Es raro ver una demo con música de guitarra y trompetas. Cuesta describir con palabras el tipo de música de Arte, pero creedme, vale la pena escucharla.

Parece mentira, pero en una producción en la que destacan todos los aspectos cómo en "Arte," se hace difícil el comentarla y transmitir a los lectores toda su fuerza y espectacularidad. Así que haced el favor de conseguir esta demo ¡ya! si no la tenéis. Vale la pena, de verdad. Es una de las mejores trackmos de diferentes efectos que se han hecho últimamente. ¡Y todo ello en una máquina no AGA!



NOVEDADES

En esta sección podréis ver los próximos juegos que saldrán a la venta. Por desgracia, debido al casi inexistente mercado de videojuegos para Amiga en España, es posible que nunca veamos su comercialización en nuestro país. Aquellos que deseéis ver las novedades que podrán disfrutar nuestros vecinos europeos en cuyo país exista un sólido mercado de videojuegos para Amiga sólo debéis echar un vistazo a la siguiente lista. Ojalá sirviera esta sección para que la gente presionase a comercios y distribuidores demandando juegos para Amiga, y de nuevo tuviéramos juegos en cualquier tienda... sin embargo, de momento, podemos conseguirlos solicitándolos a los distribuidores de Amiga (MailSoft, Tu Amiga Ordinadors y Amiga Center son algunos de ellos, y nos han confirmado que importan y pueden conseguir cualquier juego de Amiga directamente del extranjero).

NOMBRE

Allen Breed 2: Tower Assault
 Banshee AGA & CD32
 Benefactor
 Bump'n'Burn
 Darkseed CD32
 Detroit
 Dragonstone
 Eifmania
 Heimdall 2
 Impossible Mission 2025 AGA
 Ishar 3
 James Pond 3
 Legacy of Sorasil CD32
 Lili Devil
 Lords of the Realm
 Out to Lunch AGA
 Ruff'n'Tumble
 Seventh Sword of Mendon
 Skeleton Krew AGA & CD32
 Super Methane Brothers
 TFX
 Top Gear 2 AGA
 Traps'n'Treasures
 Troll Islands
 UFO: Enemy Unknown
 Universe

COMPANÍA

Team 17
 Core Design
 Psygnosis
 Grandslam
 Cyberdreams
 Impressions
 Core Design
 Terramarque
 Core Design
 MicroProse
 Daze/Silmarils
 Millennium
 Gremlin
 Gremlin
 Impressions
 Mindscape
 Renegade
 Grandslam
 Core Design
 Extender
 Ocean
 Gremlin
 Krisalis
 Millennium
 MicroProse
 Core Design

TIPO

Arcade
 Arcade
 Plataformas
 Simulador Deportivo
 Aventura Gráfica
 Estrategia
 RPG
 Lucha
 RPG
 Plataformas
 RPG
 Plataformas
 RPG
 Plataformas
 RPG
 Aventura
 Estrategia
 Plataformas
 Plataformas
 RPG
 RPG
 Plataformas
 Simulador Aereo
 Simulador Deportivo
 Plataformas
 Plataformas
 Estrategia
 Aventura Gráfica

MUNDIAL '94: Fútbol hasta en la sopa

Cuando leáis este artículo, el mundial de fútbol habrá terminado. Sin embargo, y como siempre, las compañías de software aprovechan cualquier evento deportivo importante para poder hacer su agosto. Así nos encontramos ahora: el mundial terminó hace tiempo pero los juegos de fútbol y simuladores de entrenador no dejan de aparecer (y hay cuerda para rato). Veamos que novedades han aparecido o están a punto de hacerlo en el mercado europeo (a excepción del nuestro, claro):

NOMBRE

Champ. Manager E.S. Data Disk
 Club Football - The Manager
 Empire Soccer '94
 FIFA International Soccer
 International Soccer
 Kick Off 3
 On the Ball
 Real World Football
 Sensible Soccer Int. Ed.
 Sensible World of Soccer
 Wembley International Soccer
 World Cup USA '94

COMPANÍA

Domark
 The Software Bussines
 Empire
 Electronic Arts
 Zeppelin Games
 Anco
 Daze
 Infogrames
 Sensible Software
 Sensible Software
 Audiogenic
 U.S. Gold

TIPO

Simulador
 Simulador
 Simulador Deportivo
 Simulador Deportivo
 Simulador Deportivo
 Simulador Deportivo
 Simulador
 Simulador Deportivo
 Simulador Deportivo
 Simulador Deportivo
 Simulador Deportivo
 Simulador Deportivo



ENCUENTRA TU DISTRIBUIDOR AMIGA MÁS CERCANO...

DISTRIBUIDORES Y TIENDAS DE AMIGA

BARCELONA

VB SOFT S.L.
 C/ Provenza, 435, 2-3
 08025 Barcelona
 Tlf. 93 456 15 45 Fax. 93 456 15 45

AMIGA CENTER
 Argullés, 127
 08016 Barcelona
 Tlf. 93 276 38 06 Fax. 93 276 30 80

MAIL SOFT
 C/Pau Claris 106
 Barcelona
 Tlf. 412 63 10

TU AMIGA ORDINADORS
 Plaza Pedro IV, 3 local 9
 08120 La Ilagosta (Barcelona)
 Tlf/Fax. 93 560 76 12

METALSOFT PD
 Apdo. 673
 08220 Terrasa (Barcelona)
 Tlf. 93 783 13 31

RADIO DEFOREST
 Viladomat, 105
 08015 BARCELONA
 Tlf. 93 423 72 29

CIMEX ELECTRONICA
 Sepulveda, 167
 08011 BARCELONA
 Tlf. 93 454 70 42

SALVADOR SERRA
 P. de Gracia, 22
 08007 BARCELONA
 Tlf. 318 04 78

PALENCIA

PIXELSOFT S.L.
 Felipe II, 3 BIS
 34004 PALENCIA
 Tlf. 979 71 27 00

MADRID

CENTRO INFORMÁTICO BOADILLA (CIB)
 Convento, 6 1-4
 28660 Boadilla del Monte, MADRID
 Tlf. 91 632 27 65 / 632 23 75 Fax. 91 632 10 99

DAT
 Ingenieros S.L.
 C/ General Yagüe, 15, 3A
 28440 Guadarrama - MADRID
 Tlf. 91 854 11 67



VIZCAYA

C.R.E. Electrónica
 San Francisco, 85
 48003 Bilbao, VIZCAYA
 Tlf. 94 444 98 84 Fax. 94 444 98 84

MÁLAGA

MEMORY INFORMÁTICA S.L.
 Alamos, 1
 29012 MÁLAGA
 Tlf. 95 260 00 06 Fax. 95 221 05 66

ALICANTE

MICRO K INFORMÁTICA MKI S.L.
 Segura, 27
 03004 ALICANTE
 Tlf. 96 514 37 34 Fax. 96 514 37 34

PONTEVEDRA

GALIFRAME
 Galerías Príncipe 22, 1
 VIGO
 Tlf-Fax. 986 22 89 94

ORENSE

NORSOFT
 C/ Badaya, 4
 32003 Orense
 Tlf. 988 24 90 46 Fax. 988 23 42 07

GUIPUZCUA

DONOSTI FRAME
 Servicios Informáticos
 Avda. de Madrid, 15
 20011 San Sebastián
 Tlf. 943 42 07 45 Fax. 943 42 45 88

VALENCIA

INVISION S.L.
 C/ Salamanca, 53
 46005 VALENCIA
 Tlf. 93 395 02 43/44 Fax. 96 395 02 44

NOTA: Amiga.InFo no ha podido contactar con algunos de los anteriores distribuidores, y ante la duda, hemos optado por publicar la dirección. Si eres un distribuidor de Amiga y no figuras como tal o tus datos no son correctos, ponte en contacto con la redacción y te incluiremos gratuitamente: AMIGA.InFo APDO. 166 08750 Molins de Rei BARCELONA



LLAMA...INFÓRMATE...
 ELLOS TIENEN LO QUE BUSCAS...
¡PÍDELES TU AMIGA.InFo!

¡Recibe el próximo número de AMIGA.InFo en tu domicilio!

Deseo recibir:

☐ Revista AMIGA.InFo + Discos de Portada: (1 número + 2 disquetes) por 700 pts. + 300 pts. de gastos de envío.

Nombre _____ Edad _____
 Apellidos _____
 Profesión _____ Empresa _____
 Dirección _____
 DNI _____ Localidad _____
 Provincia _____ CP _____
 Teléfono (indicar prefijo) _____

Forma de Pago:

☐ Ingreso al nº de cuenta: 2100-0046-21-0103285477 (La Caixa u otras)
☐ Contra reembolso por correos. (Añadir 225 pts.).

Enviar esta página o una fotocopia a:

AMIGA.InFo (Ref: Reserva)
Apartado de Correos 166
08750 Molins de Rei
Barcelona

BUSES, CPU's, MEMORIAS, E/S, DMA's y ¿ALGO MAS?



¿Alguna vez has sentido curiosidad por saber qué se esconde dentro de la carcasa de tu Amiga? ¿Sabes qué es un procesador, memoria CHIP o un CIA? Hummm...Puede que hayas

oído hablar alguna vez de ellos, pero...¿y de canales DMA o de custom chips VLSI? ¿No? En esta sección, F. Martín nos comentará cada mes todo lo relacionado con el 'hardware', es decir, los componentes electrónicos del AMIGA: veremos la función y la programación individual de cada uno y cómo interactúan para conseguir realizar las tareas internas de un ordenador. Estudiaremos la arquitectura del Amiga.

Actualmente, muchos son los usuarios de ordenadores Amiga de habla hispana que no poseen información alguna de este magnífico ordenador. Esta sección pretende, en la medida de lo posible, llenar ese vacío conocido por todos los que han buscado y no han encontrado información suficiente acerca del 'hardware Amiga'.

Tanto los programadores (expertos o no) como los usuarios que tienen curiosidad por saber algo más sobre la máquina que tienen en las manos, pueden utilizar esta sección como guía de consulta o referencia sobre temas concretos de los componentes que integran y forman el Amiga.

La primera impresión que tuve al trabajar con un Amiga fue la de sorpresa, debida a la facilidad de manejo y a su admirable capacidad gráfica y de sonido. El entorno multitarea a través de pantallas y ventanas ofrece al usuario unas prestaciones mayores que en otros sistemas, y ésto es debido al diseño particular y personalizado que los ingenieros aplicaron al crear en 1985 el Amiga. Desde esa fecha, han ido apareciendo versiones mejoradas del hardware y del sistema operativo, que añadían nuevas capacidades (sobre todo gráficas) hasta llegar actualmente a los modelos con arquitectura de 32 bits y chips AGA.

Frente a esta diversidad de modelos de Amiga que existen, y aunque son compatibles entre ellos, he creído conveniente explicar cada uno de los bloques de forma independiente al modelo concreto utilizado por el lector. Esto no implica que lectores con un A1200 o A4000 se vean discriminados, todo lo contrario, para estos modelos recientes he prestado especial atención, comentando las diferencias con los modelos básicos (A1000, A500, A600, A2000,

A3000) y explicando todas las nuevas capacidades gráficas cuando sea necesario.

De igual forma cuando me planteé que nivel debía utilizar para poder explicar las funciones y la forma de programar cada uno de los chips del Amiga, tuve que establecer unos 'límites' para no convertir los artículos en documentos de 2000 páginas ilegibles. Este nivel está situado entre el sistema operativo (justo debajo) y los registros (posiciones de memoria) que cada dispositivo posee para su control y programación. No he explicado el sistema operativo, ya que es tema de otra sección que podéis encontrar en esta misma revista, ni tampoco la parte electrónica, que pertenece también a otra sección.

Desde este punto de vista, se podría abusar y decir que únicamente se ve el Amiga como unos conjuntos de registros, cada uno con una determinada función dentro del sistema. Cada chip es controlado por uno o varios de estos registros, y será tarea nuestra programarlos para que den los resultados que deseemos.

Comentaré algunas ventajas de esta forma de acceso mediante registros:

- ✗ El acceso directo a los registros suele ofrecer una visión al programador de las posibilidades reales de cada componente del Amiga.
- ✗ Se obtiene mayor eficiencia y código más compacto en los programas. La velocidad de ejecución es prácticamente la máxima que se puede obtener del Amiga.
- ✗ Se puede acceder fácilmente a dispositivos hardware que de otra forma se convertiría en una tarea complicada, sino imposible.

Sin embargo, la programación directa del hardware trae algunos problemas en un sistema

multitarea, donde pueden coexistir varios programas ejecutándose simultáneamente que utilizan los mismos recursos (los mismos componentes del sistema). Este tipo de acceso puede provocar un caos en el sistema operativo si no se presta atención y se modifica el contenido de los registros de forma descuidada. En este punto es importante diferenciar la forma más conveniente de programar:

✓ Si nuestro programa ha de correr en multitarea con el sistema operativo (workbench y demás) u otros programas, no es nada recomendable utilizar directamente los registros (es más, está prohibido por Commodore). Una forma de estar totalmente seguros de que nuestro programa no va a interferir con otros, es accediendo al hardware indirectamente a través de llamadas al sistema operativo. Sin embargo, de esta forma se pierde velocidad y se limitan las posibilidades, aunque es la forma más utilizada.

✓ Por el contrario, si nuestro programa ha de tomar control absoluto del ordenador, sin necesidad de utilizar el sistema operativo, la forma más conveniente es la programación directa del hardware.

He comentado la forma de programar, pero no el lenguaje de programación. Aquí, de nuevo, tuve que elegir entre un lenguaje como Modula II, Pascal, C o E y el lenguaje ensamblador. Los primeros no ofrecen posibilidades a la hora de programar y acceder a niveles bajos del sistema, a excepción de C y E, que sí están preparados para trabajar con direcciones y datos a nivel de bits. Pero el código resultante de C o E no es tan eficiente y compacto como el resultante de programar en código ensamblador (¡A parte de que es

necesario tener el compilador!). Por esta razón, todos los ejemplos contenidos en esta sección están en ensamblador, y además ofrecemos al lector no sólo abundantes programas, sino un entorno donde se puedan modificar y utilizar los ejemplos ya tecleados, incluidos en los discos de la revista, junto con diversas utilidades como un ensamblador y editor de dominio público. De esta forma, el lector está preparado para programar inmediatamente, ya sean los ejemplos o sus propios programas.

Mes a mes iremos recorriendo y explicando detalladamente cada parte, mostrando las posibilidades que cada chip o circuito integrado tiene y la forma de programarlo. Pero para iniciar este interesante viaje, el primer paso es coger el avión y dar un "vuelo de reconocimiento" sobre conceptos básicos y componentes relacionados con los ordenadores. Así conoceremos el que será nuestro lugar de trabajo habitual... ¡y qué mejor para centrarnos en el tema que 'diseñar a "grosso modo" un ordenador!

'DISEÑEMOS' UN ORDENADOR

Imaginemos que por un rato somos ingenieros de una empresa fabricante de ordenadores y que hemos recibido el proyecto de diseñar un ordenador dentro de la categoría de ordenadores personales. Nos situamos en una sala de reuniones de nuestra empresa (año 1985), donde el equipo de ingenieros está discutiendo en una primera reunión que componentes, partes y características han de formar nuestro ordenador. No están entrando en detalles, sino estableciendo criterios y bases generales de una primera fase de diseño. Veamos lo que podemos aprender de ellos:

- En primer lugar, necesitamos un circuito que pueda entender las instrucciones de un lenguaje de programación, ejecute programas y procese la información de una forma rápida y sencilla, es decir, hay que buscar un microprocesador. Podemos diseñarlo nosotros, pero sería costoso y lento, y por otra parte, en el mercado actual parece que tienen mucho éxito los microprocesadores de la empresa Intel y Motorola. Ya veremos más adelante cuál de ellos se adapta mejor a nuestro esquema.

- También necesitamos un circuito que nos permita almacenar los datos y los programas, pero ha de ser rápido puesto que los datos han de ser accedidos continuamente por el procesador y otros componentes, y además de forma aleatoria y directa, es decir, necesitamos una memoria DRAM (Dynamic Random Access Memory). Estas memorias están divididas en casillas donde se almacena individualmente la información (de tamaño 8 bits). Cada casilla tiene su dirección que la identifica, y de esta forma, cuando queramos acceder al contenido de una de ellas, únicamente tenemos que entregarle a este circuito una dirección (un valor numérico) e indicarle si queremos leer o escribir en esa casilla. En caso de lectura, nos entregará inmediatamente el contenido de la casilla, en caso de escritura, nosotros le proporcionaremos también el dato a escribir. El procesador, entre otros, accederá a la memoria

generando direcciones y leyendo o escribiendo en el contenido de ellas. Hay varios tipos de DRAMs en el mercado, pero la que necesitamos nosotros es la más utilizada y económica (Tipo DIP: Dual In line Pin).

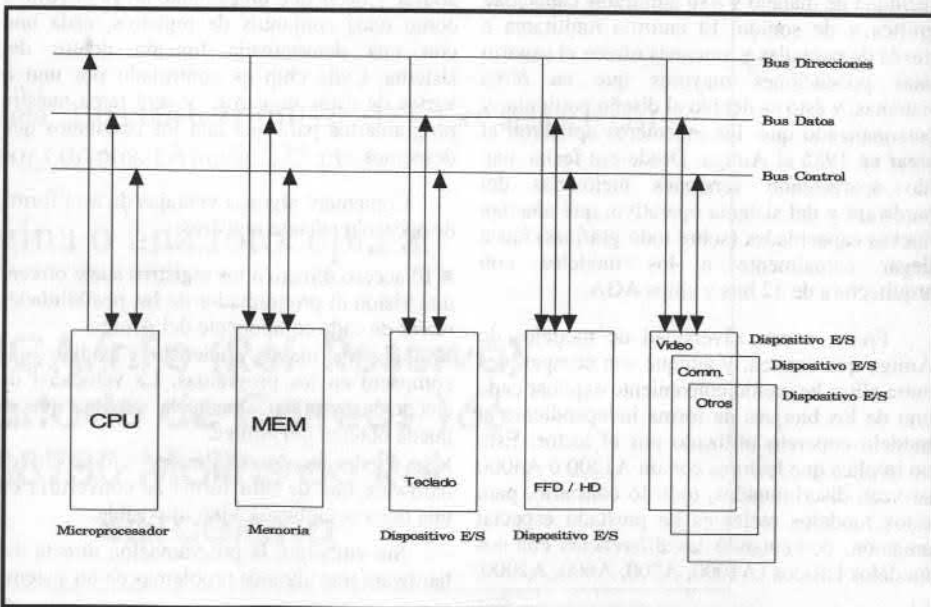
Por otro lado, ahora que tenemos un procesador y una memoria, necesitamos algún medio para introducir la información, es decir, dispositivos de Entrada/Salida (E/S) de datos como puedan ser un teclado, ratón, pantalla, unidad de disco, disco duro, puertos externos serie y paralelo...y sus respectivos circuitos que los controlen y permitan el acceso a ellos.

Con estos dispositivos nos aseguramos que la información va a estar almacenada de forma permanente y segura, y que podrá ser recuperada y modificada con facilidad (aunque con menos velocidad que la memoria DRAM). También permitimos que se introduzcan datos de forma directa por el usuario, utilizando un teclado y un ratón. Ya consideraremos si utilizamos controladores de dispositivos que existan en el mercado, o los diseñamos nosotros.

Ahora estamos preparados para esbozar un primer esquema donde aparezcan estos tres bloques: el procesador, la memoria, y los dispositivos de Entrada/Salida. Sin embargo, todavía no hemos comentado un punto importante: la forma de interconectar estos módulos entre sí. Podemos utilizar un sistema de conexión típico que utiliza líneas eléctricas (buses) para comunicar los diferentes bloques. La idea es que en la placa donde residen los componentes conectamos un bloque con otro a partir de 'cables' o 'pistas' trazadas en la propia placa. Hay que tener cuidado en este punto, puesto que cuantas más líneas posea nuestro diseño, más difícil y caro será fabricarlo. Por lo tanto, no podemos dedicar buses diferentes para comunicar entre ellos la memoria, procesador y dispositivos E/S.

Actualmente existe una técnica que utiliza únicamente tres buses y permite conectar un número elevado de componentes entre sí de

FIGURA 1. Muestra de la configuración básica que los ordenadores suelen tener. Se observan los tres buses y conectados a ellos la memoria, la cpu, y la Entrada/Salida.



forma simple y económica. Veamos que tipo de buses son:

□ **Bus de datos:** Este bus proporciona un camino para mover los datos entre los módulos que forman el sistema. Por ejemplo, la memoria obtiene de este bus los datos que deposita el microprocesador en el mismo bus, y viceversa. Cuantas más líneas posea este bus, mayor será la cantidad de información que podrá ser transferida simultáneamente (normalmente suelen ser de 16 ó 32 bits)

□ **Bus de direcciones:** Es usado para determinar el destino y origen de los datos colocados en el bus de datos. Por ejemplo, si la CPU quiere leer un dato de memoria, coloca su correspondiente dirección en este bus y avisa a la memoria de este hecho. Después, la memoria le colocará el dato asociado a esa dirección en el bus de datos y la CPU lo leerá. La anchura en bits de este bus limita la cantidad de direcciones accesibles. (Por ejemplo, si es de 24 ó 32 bits, se consiguen 2^{24} ó 2^{32} direcciones diferentes)

□ **Bus de control:** Todas las señales de control entre los diversos componentes, como por ejemplo señales de sincronización, permiso de Lectura/Escritura, activación, relojes... se reúnen en este bus.

En un principio, parece conveniente que utilicemos sólo buses de este tipo, puesto que simplificará en gran medida el diseño. Ahora que sabemos como conectaremos los bloques, hagamos un diagrama en papel que nos ayude a tener una visión gráfica de nuestro primer diseño (Mostrado en la figura 1). Se puede intuir de forma aproximada el funcionamiento básico de este esquema:

El procesador se encarga de leer y ejecutar los programas de memoria para obtener unos resultados, que serán enviados a los dispositivos de E/S convenientemente o almacenados en la memoria. Para ello, se emplean los buses como medio de transferencia de la información. Es importante observar que todos los componentes comparten los mismos buses y por lo tanto,

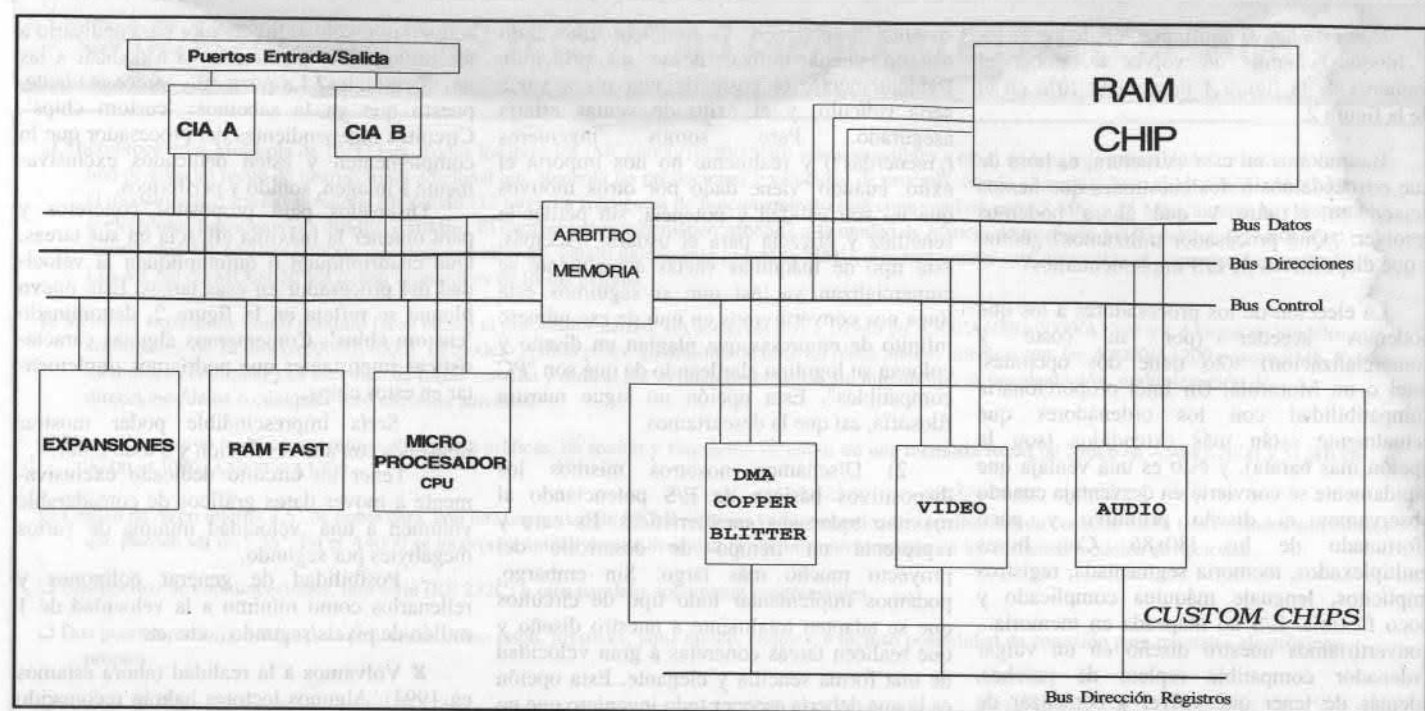


FIGURA 2. Se observa la principal diferencia entre el esquema de la fig.2: Aparecen buses locales al procesador y a los dispositivos de E/S separados por un árbitro de memoria. El bloque denominado "custom chips" es el más importante puesto que permite realizar todas las tareas gráficas y de sonido, así como de E/S de forma independiente a la CPU.

nos encontramos con una limitación seria: durante el acceso de dos componentes al bus (uno para recibir y otro para enviar) el resto de los bloques han de desconectarse para no colisionar con los datos que están siendo transmitidos. Afortunadamente, los circuitos están preparados para este tipo de desconexiones. Cuando no son utilizados, automáticamente entran en un estado denominado de 'alta impedancia', muy similar a lo que podría ser extraer o desconectar físicamente el componente del bus.

Sin embargo, el siguiente ejemplo nos descubre otro problema en la configuración de la figura 1:

Imaginemos que el microprocesador necesita datos que estén en la memoria. El protocolo típico que suele seguir para conseguirlos es apropiarse de los buses si están libres...pero ¿y si están ocupados por otro componente? Podría darse el caso de que un dispositivo de E/S estuviera transfiriendo directamente datos de disco a memoria. El microprocesador no tiene más que dos soluciones: pararse y esperar o robarle los buses al dispositivo. Esto significa que o bien nuestro programa deja de ejecutarse (!!!) o el dispositivo deja de leer de disco. En un ordenador, los accesos a disco, o más todavía a memoria, son continuos y muy frecuentes, por lo tanto la ejecución de nuestro programa será interrumpida constantemente, disminuyendo el rendimiento final. Esto hay que evitarlo.

Este problema se puede extender a cualquier combinación posible de acceso a los buses por parte de los distintos bloques, así que debemos solucionarlo de alguna forma. Reflexionando sobre el tema nos podríamos preguntar si es más conveniente buscarle una solución complicada o simplemente establecer, por ejemplo, una prioridad de acceso a los

buses, dando más privilegio a los componentes que tengan accesos frecuentes y necesiten tiempos críticos de respuesta, como puedan ser el procesador y la memoria. Esta solución no es la más óptima. Hay componentes que la mayoría del tiempo estarían parados a la espera de que se les ceda los buses (por ejemplo los dispositivos de E/S) e incluso una vez que los posean, puede ocurrir que un componente con mayor privilegio ¡se los robe!

Quizás, en lugar de utilizar prioridades, podríamos asignar a cada componente un determinado tiempo de acceso a los buses, durante el cual los demás permanecen desactivados. Una vez se consuma el tiempo asignado se cederán los buses a otro componente que esté en espera. De esta forma, todos los bloques tendrán oportunidad de transferir información. Ok, parece lógico, pero ¿quién determina el tiempo que ha de estar cada uno de los componentes o decide que se ha excedido el tiempo de posesión? Los propios componentes no pueden decidir por sí solos, así que es necesario un circuito adicional que coordine y controle los accesos: necesitamos un árbitro de buses. Esta solución nos atrae, pero enseguida descubrimos que para poder repartir los buses en el tiempo los componentes conectados a ellos han de estar preparados...han de ser síncronos...y este tipo de componentes son caros y complicados de diseñar. De nuevo no hemos dado con la solución óptima.

Parece evidente que utilizando un único sistema de buses el problema no tiene una solución rápida y sencilla. ¿Y si añadimos más buses? Podemos obtener un compromiso entre número de buses, el coste y el rendimiento conseguido si dedicamos exclusivamente tres buses del mismo tipo que los anteriores a la memoria y procesador, y otros tres a los dispositivos de E/S y memoria. De esta forma

todos trabajarían al 100% de su velocidad ya que no existirían colisiones.

Eso sí, la memoria hay que dividirla en dos bloques totalmente diferentes, puesto que si es compartida de nuevo aparecen problemas de colisión. Pero si tenemos dos memorias separadas y no están compartidas por los componentes, ¿cómo consigue el procesador obtener los datos que un dispositivo de E/S ha dejado en la memoria que le corresponde?

Tenemos que unir los buses de las dos partes para que en caso necesario, el procesador pueda acceder a la memoria asignada a los dispositivos de E/S. (Para estos últimos no es necesario que accedan a la del procesador o volveríamos al principio). La unión de los buses podemos realizarla a través de un árbitro de memoria que permita de forma opcional (y cuando convenga) acceder al procesador a los dispositivos de E/S y su memoria. OK!!! Parece que hemos dado con la solución, veamos, a priori, que dos ventajas clave tiene:

✓ Permite en el mejor de los casos trabajar a la máxima velocidad que los componentes puedan dar. Esto es importantísimo. En el peor de los casos, el único que verá disminuido mínimamente su rendimiento será el procesador, puesto que esperará a que el árbitro le ceda los buses.

✓ Mediante esta configuración, podemos dar más independencia a las actividades que los dispositivos de E/S han de realizar. Se descarga al procesador de tareas que no son propias de él, como la lectura del teclado, acceso a pantalla o carga directa en memoria de datos que proceden de una unidad de disco (DMA: Acceso directo a memoria). Incluso podemos incluir algún coprocesador gráfico que ejecute sus propios programas...

Con esta nueva configuración de los buses y bloques, hemos de volver a retocar el esquema de la figura 1 para convertirlo en el de la figura 2.

Basándonos en esta estructura, es hora de que retrocedamos a dos cuestiones que hemos dejado en el aire y que ahora podemos resolver: ¿Qué procesador utilizamos? ¿cómo y qué dispositivos de E/S implementamos?

La elección de los procesadores a los que podemos acceder (por su coste y comercialización) sólo tiene dos opciones: Intel o un Motorola. Un Intel proporcionaría compatibilidad con los ordenadores que actualmente están más extendidos (son la opción más barata), y esto es una ventaja que rápidamente se convierte en desventaja cuando observamos el diseño primitivo y poco afortunado de los i80x86. Con buses multiplexados, memoria segmentada, registros implícitos, lenguaje máquina complicado y poco flexible, E/S no mapeada en memoria... convertiríamos nuestro diseño en un vulgar ordenador compatible repleto de parches, además de tener que volver a comenzar de nuevo todo el proceso de diseño con más problemas que soluciones.

Veamos que ofrece un Motorola 680x0:

- Lenguaje máquina potente y completo. Libertad total en el uso de los registros internos y programación sencilla.
- Buses sin multiplexar, es decir, puede direccionar directamente un espacio continuo de memoria de como mínimo 16 Mbytes.
- Los dispositivos de E/S están mapeados en memoria, y se accede a ellos como una posición de memoria se tratara.
- Arquitectura interna de 32 bits.
- Es posible conectarle circuitos periféricos tanto de forma síncrona como asíncrona.
- Existen varios modelos que funcionan a distintas velocidades y contienen mejoras internas que aumentan el rendimiento. Todos los modelos son compatibles en el lenguaje máquina de forma ascendente.
- Facilidades para ser integrado en un sistema multitarea.

Este procesador se adapta totalmente a los requisitos de nuestro diseño, así que, será la mejor elección.

En cuanto a los dispositivos de E/S, hemos de discutir que posición tomamos:

1) Dejamos unos puertos con zócalos preparados para permitir conectar directamente a los buses estos dispositivos y que otros fabricantes se dediquen a diseñarlos, como puedan ser controladoras de vídeo, disco, disco duro, teclado, puerto serie, paralelo, sonido... Es decir, fabricamos una placa vacía donde sólo estén implementados los buses, el procesador, la memoria y algún circuito adicional secundario. Esta es la posición más cómoda y desde luego la más económica. Pero dejar aspectos tan importantes en manos de terceros puede ocasionar la aparición de multitud de tarjetas incompatibles entre sí, que puede producir un caos en el usuario frente a esa diversidad. Los programadores tendrían que estar constantemente modificando sus aplicaciones para abarcar al máximo todo tipo de tarjetas diferentes. No conseguiríamos un

sistema homogéneo y completo que todo usuario pueda utilizar desde un principio. Evidentemente, el coste de una placa vacía sería ridículo, y el éxito de ventas estaría asegurado. Pero somos ingenieros (¿recuerdas?) y realmente no nos importa el éxito, cuando viene dado por otros motivos que no son calidad y potencia, sin perder la sencillez y eficacia para el usuario. Además, este tipo de máquinas vacías actualmente se comercializan ya, así que si seguimos esta línea nos convertiremos en uno de ese número infinito de empresas que plagian un diseño y colocan su logotipo alardeando de que son "PC compatibles". Esta opción no sigue nuestra filosofía, así que la descartamos.

2) Diseñamos nosotros mismos los dispositivos básicos de E/S potenciando al máximo todas sus características. Es caro y representa un tiempo de desarrollo del proyecto mucho más largo. Sin embargo, podemos implementar todo tipo de circuitos que se adapten totalmente a nuestro diseño y que realicen tareas concretas a gran velocidad de una forma sencilla y elegante. Esta opción es la que debería escoger todo ingeniero que no esté presionado por motivos económicos o temporales. Total libertad para crear chips especiales dedicados a aspectos como gráficos, vídeo, sonido, puertos de comunicación...etc.

Crearemos 'custom chips', es decir, chips especiales y personalizados para el diseño de nuestro proyecto. Pero... ¿por qué no empleamos el procesador en tareas de imagen, sonido y periféricos?

Si aumentamos su velocidad, por ejemplo, a 16, 33 o mejor aún, a 66 Mhz, siempre nos sobrará tiempo de procesador para ejecutar otras tareas como presentar gráficos, transferencias de datos, acceso a periféricos... ¡NO! No podemos basar nuestro diseño únicamente en el procesador, y menos aún confiar en que el aumento de la frecuencia de reloj nos salvará en un futuro. Además, a medida que aumentamos la velocidad también aumenta la potencia consumida, y ¡¡¡ el calor disipado!!!. Los sistemas de refrigeración que habría que emplear serían muy voluminosos y caros. Un procesador funcionando a 66 Mhz podría tener serios problemas en estos y otros muchos aspectos que, aunque si es posible solucionarlos, no es el camino que debemos seguir...

Por otro lado, tareas como acceder a pantalla, mover gráficos, o transferir datos a disco ocuparían demasiado tiempo para un procesador, puesto que también ha de ejecutar programas y realizar funciones propias de él. Estaría excesivamente sobrecargado como para soportar multitarea real de una forma decente. Imaginamos que ha de mover un bloque de datos de 4 Mbytes, evidentemente que lo puede hacer, pero no de la forma más eficiente (un chip diseñado para mover bloques puede alcanzar hasta 8 megas/seg).

El procesador lo dedicaremos exclusivamente a obtener la máxima velocidad de ejecución de programas. Como consecuencia, podremos implementar un mecanismo que nos permita repartir el tiempo de procesador entre diferentes programas, es decir, multitarea. Pero

si queremos sólo el procesador para dedicarlo a los programas... ¿Quién se va a dedicar a las tareas restantes? La respuesta parece evidente, puesto que ya la sabemos: 'custom chips'. Circuitos independientes del procesador que lo complementen y estén dedicados exclusivamente a imagen, sonido y periféricos.

Diseñados para propósitos concretos y para obtener la máxima eficacia en sus tareas. Que cuadruplicen o quintuplicen la velocidad del procesador en esas tareas. Este nuevo bloque se refleja en la figura 2, denominado 'custom chips'. Comentemos algunas características importantes que podríamos implementar en estos chips:

- Sería imprescindible poder mostrar imágenes con alta resolución y a todo color.
- Tener un circuito dedicado exclusivamente a mover datos gráficos de considerable volumen a una velocidad mínima de varios megabytes por segundo.
- Posibilidad de generar polígonos y rellenarlos como mínimo a la velocidad de 1 millón de pixels/segundo...etc...etc

✕ Volvamos a la realidad (ahora estamos en 1994). Algunos lectores habrán reconocido en el primer esquema la estructura típica de un PC (IBM o compatible) y también habrán observado que las soluciones que íbamos descartando ¡son las que han utilizado estos ordenadores!. Desde aproximadamente los 70 hasta hoy día, los principios en que se basan los PCs no han cambiado. Únicamente han evolucionado en un aspecto: Aumentan la potencia de sus procesadores con diseños complicados a base de elevar la frecuencia de funcionamiento.

El diseño de la figura 2 se creó aproximadamente en 1985. Por aquel entonces, apareció en el mercado el Amiga 1000. Hoy día, el esquema ha evolucionado y mejorado en muchos aspectos, pero sigue manteniendo la misma filosofía: "Procesador-Memoria-Buses Dedicados-Custom Chips". Llegados a este punto, es conveniente que especifiquemos las características técnicas de los Amiga actuales, para que todos podamos tener una misma visión global aproximada de sus capacidades.

ESPECIFICACIONES TÉCNICAS

El mundo Amiga está compuesto por diferentes modelos, aunque siempre se ha intentado conservar la compatibilidad entre todos ellos, compartiendo la misma base hardware, es decir, las mismas capacidades de sonido y gráficos. A cada modelo nuevo desarrollado se le han implementado todas las características del anterior, y además se le han añadido mejoras, como ha ocurrido entre los modelos NO AGA (A500, A2000, A3000...) y AGA (A1200, A4000).

Evidentemente existen diferencias, algunas de ellas muy importantes, como son los procesadores que llevan incorporados (M68000 /M68020 /M68030 /M68040) o la capacidad de expansión interna y externa, es decir, máxima cantidad de memoria permitida, el número de puertos donde colocar tarjetas internas, controladoras de todo tipo, genlocks, modems...etc. En las siguientes tablas mostramos los componentes y características que todos los Amiga incorporan.

CARACTERÍSTICAS TÉCNICAS GENERALES DE LOS AMIGA

- Microprocesador Motorola de 16/32 bits. Todos los modelos (MC68000/10/20/30/40) son compatibles a nivel de lenguaje ensamblador, ofreciendo cada uno de ellos un juego de instrucciones adicional que aumenta las prestaciones y velocidad de proceso, siendo el modelo base el MC68000 (16/32 bits) y el modelo actual más avanzado el MC68040 (32 bits). La frecuencia de funcionamiento está comprendida entre 7 Mhz y 25 Mhz, aunque hay disponibles aceleradoras funcionando a 40Mhz y 50Mhz. El coprocesador matemático M68882 y la unidad de manejo de memoria MMU están integrados en el mismo chip MC68040 (A4000), mientras que en el MC68030 únicamente lo está la MMU, y externamente está disponible el coprocesador matemático. Los MC68000/MC68020 los tienen disponibles de forma externa.
- Memoria expandible como máximo (si se utiliza el procesador nativo, sin aceleradoras) a 10 Mb en A500/A600/A2000/A1200 y a 4 Gigs en modelos con arquitectura de 32 bits (A3000/A4000). El modelo A4000 posee un mínimo de 6Mb en placa madre, mientras que los A3000/A1200 poseen 2Mb. Esta memoria es continua, y es accedida de forma sencilla y directa por el microprocesador, sin ningún tipo de segmentación, multiplexación de direcciones/datos o cualquier otro esquema adicional.
- Sistema operativo multitarea y rutinas de soporte gráficos, de sonido y animación incluidos en una memoria ROM de 256kb en A500/A2000 y 512kb en A600, A3000, A4000 y A1200.
- Unidad de disco interna 3.5" de doble cara, con una capacidad de 880kb en disco ya formateado, con posibilidad de conectar hasta tres unidades adicionales que pueden ser de 5.25". En el A4000 se incorpora unidad de alta densidad (1.76 mb), mientras que en los restantes modelos es opcional.
- Dos puertos de comunicaciones, uno serie (RS 232C) y otro paralelo, totalmente programables.
- Dos puertos reconfigurables para periféricos como ratón, joysticks, lápiz óptico, paddles, e incluso posibilidad de conexión para montajes electrónicos propios.
- Puertos de vídeo con salida simultánea de vídeo compuesto, analógica o digital RGB (Red Green Blue).
- Puertos para la salida de sonido estéreo de los cuatro canales de audio.
- Controladora interna de disco duro tipo SCSI (A3000) o IDE (A600 / A1200 / A4000). Opcionalmente existen controladoras de tipo SCSI II.
- Teclado profesional con keypad numérico, diez teclas de función y teclas de cursor. Cada teclado está configurado según el país, aunque existe posibilidad de reconfigurarlo por software.
- Puerto compatible para tarjetas PCMCIA II, como ampliaciones de memoria DRAM y SRAM, modems, discos duros, CDROMs...etc (Sólo en modelos A600/A1200)
- Puerto interno donde están disponibles todas las señales de los buses y del microprocesador para posibles tarjetas con ampliaciones o cambios del microprocesador no necesariamente Motorola. También pueden ser, por ejemplo, Intel y dar compatibilidad total con MSDOS, obteniendo dos sistemas con programas en multitarea.

AMIGA CUSTOM CHIPS

Todas las especificaciones anteriores suelen ser comunes en todo tipo de ordenadores, sin embargo, hay que prestar especial atención a los denominados 'custom chips' diseñados y personalizados exclusivamente para trabajar en un entorno multitarea. Estos, diferencian al Amiga de todos los demás ordenadores, consiguiendo un alto rendimiento del sistema gracias a ellos.

Estos tres chips (llamados Agnus o Alice, Paula, Denise o Lisa) están diseñados para desarrollar tareas conjuntamente en temas como vídeo, sonido, acceso directo a memoria (DMA) o gráficos.

En este punto, hay que diferenciar sus versiones, puesto que Agnus y Denise son incorporados en A500/A2000 siendo la primera versión. Después, se mejoraron algunas características y aparecieron los

ECS (Enhanced Chip Set) en A500+/A600/A3000, para por último, llegar a los AGA (Advanced Graphics Architecture, también denominados AA 'Advanced Amiga chip set'), donde Agnus pasa a ser Alice y Denise se renombra como Lisa, desarrollándose destacadas mejoras gráficas, y una arquitectura de 32 bits total del sistema en A1200 y A4000. A pesar de que los chips AGA están muy diferenciados de sus predecesores, mantienen total compatibilidad con ellos. El chip Paula se ha mantenido casi sin cambios en esta evolución.

A continuación se muestran las funciones principales y las posibilidades que los custom chips nos proporcionan, distinguiendo cuando sea necesario las diferencias entre los modelos de Amiga:

- Gráficos de alta resolución generados a partir de bitplanos, con capacidad para soportar los dos estándares de vídeo PAL y NTSC.
 - ⇒ En los A500/A2000 se usan normalmente resoluciones de:
 - 320x256 no entrelazado
 - 320x512 entrelazado con 32 colores en pantalla
 - 640x256 no entrelazado
 - 640x512 entrelazado en pantallas con 16 colores.
 - Existen modos de vídeo adicionales que permiten ver hasta 4096 colores simultáneos en pantalla (HAM6) o conseguir mayores resoluciones como por ejemplo pantalla completa o "overscan"
 - ⇒ En los A600/A500+/A3000 se tienen todas las resoluciones del punto anterior, y además, modos de 1280x512 y 800x600 (16 colores máximo)
 - ⇒ En los A1200/A4000 se obtienen todas las anteriores resoluciones, y además modos programables entre un rango de

AMIGA CUSTOM CHIPS

resoluciones de 320x200 a 1280x1024 puntos (con frecuencias de rastreo horizontal y vertical también programable) con 256 colores en pantalla de una paleta de 16.7 millones de colores. Con un monitor de tipo A2024 se obtiene una resolución de 1024x1024 puntos. Existe un modo gráfico denominado HAM8 donde se dispone, de forma simultánea, de los 262.144 (2^{18}) colores en pantalla y en cualquier resolución.

- Un coprocesador de pantalla (denominado Copper y localizado en Agnus/Alice) que permite cambiar los valores de los registros de otros custom chips, en sincronización con el haz de electrones del monitor. Esto permite efectos especiales como cambiar la paleta de colores a partir de cierta coordenada de pantalla, u obtener varias pantallas con diferentes resoluciones, colores y demás atributos, o generar una interrupción al M680x0 sincronizada con el haz de electrones... El coprocesador puede trabajar varias veces por pantalla, en cualquier coordenada de una línea determinada o durante el intervalo en el que el haz desaparece por el final de la pantalla y aparece de nuevo por el principio (blanking interval). El coprocesador puede acceder directamente a todos los registros de los otros custom chips dejando libre al microprocesador M680x0 para otras tareas.

- Organización de los colores en bancos de registros donde se almacenan sus valores. Para cada punto (pixel) de pantalla, la combinación de los bitplanos correspondientes produce un número que selecciona (indexa) un determinado registro de color dentro del banco.

- ⇒ 32 registros de 12 bits, divididos en 4 bits para el rojo, 4 para el verde y 4 para el azul. Estos cuatro bits indican la intensidad de los tres colores base para formar una combinación de ellos y mostrarla en pantalla. Este sistema permite obtener una paleta de 4096 colores (2^4 Red $\times$ 2^4 Green $\times$ 2^4 Blue = 2^{12})
- ⇒ En los modelos A1200 y A4000, el ancho de los registros anteriores es ampliado a 25 bits, divididos en 8 bits para rojo, 8 para el verde y otros 8 para el azul, y para completar, uno para genlock. La combinación de estos valores de intensidad produce un nuevo color de una paleta de 16.777.216 colores ($2^8 \times 2^8 \times 2^8 = 2^{24}$). De igual forma, se han añadido registros, formando un total de 256. Estos están divididos en 8 bancos de 32 registros, para mantener la compatibilidad.

- Objetos gráficos denominados 'sprites' dirigidos totalmente por el hardware (Denise/Lisa), obteniéndose gran velocidad y facilidad de movimiento. Estos sprites son independientes del fondo (background) de la pantalla, ya que poseen su propio mapa de bits y registros de color y pueden aparecer sobre o debajo de pantalla sin depender de su modo gráfico y sus colores. A su vez, cada sprite es independiente de los otros, ya que se utilizan 8 canales DMA. Después de producir la última línea de la imagen de un sprite, un canal DMA puede ser reusado para producir otra imagen de otro sprite, en cualquier lugar de la pantalla (con al menos una línea horizontal de diferencia entre cada nuevo uso).

Existe una prioridad que puede ser dinámicamente controlada para cada objeto e incluso, la posibilidad de detectar colisiones. Es decir, el sistema puede controlar y modificar (a voluntad del programador) la prioridad entre los sprites y los bitplanos de fondo (playfields). De esta forma, se controla que objeto u objetos aparecen sobre o debajo del fondo en cualquier momento. Cuando diversos sprites aparecen moviéndose en pantalla, es posible que existan colisiones entre ellos, puesto que en un determinado momento dos o más objetos pueden coincidir en la misma coordenada gráfica. Estos casos pueden ser detectados automáticamente por el sistema e indicados al programador para que su programa reaccione en consecuencia a una determinada colisión. Las características gráficas de estos objetos gráficos son las siguientes:

- ⇒ 16 bits de ancho con una paleta de 15 colores (cuando los sprites son emparejados). Trabajan en modo de baja resolución con 16 pixels de ancho y un número ilimitado de pixels en altura.
- ⇒ Además, en los modelos A1200 y A4000 pueden ser mostrados en diferentes resoluciones y tener un ancho de 16, 32 ó 64 bits (pixels) e ilimitados bits en altura. Pueden usar 16 colores de una paleta de 256.

- Parte del chip Agnus (Alice en A1200/A4000) es utilizado para alojar un circuito especial empleado para mover o manipular bloques de datos a alta velocidad. Denominado 'Blitter', ha sido diseñado para recoger datos de hasta tres fuentes diferentes y combinar estos datos de 256 maneras posibles (aplicando operaciones lógicas) y opcionalmente salvar el resultado de la combinación en una zona destino. Utiliza 4 canales DMA propios e independientes entre ellos. Es posible configurarlo en un modo especial para trazar líneas en zonas rectangulares de memoria, y aplicar modelos (patterns) a estas líneas. De igual forma, puede ser empleado para el relleno de áreas de memoria, aplicando modelos si se desea. El blitter puede realizar determinadas tareas con mucha más eficiencia que el microprocesador M680x0, puesto que opera a 28 Mhz con capacidad para mover hasta cuatro megabytes por segundo y trazar líneas a la velocidad de 1 millón de pixels por segundo. La alta frecuencia de funcionamiento posibilita al programador la generación y animación en tiempo real de gráficos 2D y 3D, empleando las funciones de trazado rápido de líneas para formar complejos polígonos y la función de relleno para obtener objetos sólidos. De igual forma, es posible definir infinitos bloques gráficos ('bobs') y moverlos por pantalla rápidamente. Estos bloques tienen las mismas características que las pantallas, la misma resolución, colores y tamaño variable.

- Sonido (en chip Paula) consistente en cuatro canales digitales con programación independiente del volumen y velocidad de muestreo (sampling rate). Los canales obtienen la información vía acceso directo a memoria (DMA). Una vez inicializados, pueden emitir una forma de onda determinada sin necesidad de que el microprocesador M680x0 intervenga en ello. Dos canales son asignados a cada una de las dos salidas estéreo externas de audio. Los canales pueden trabajar entre ellos para producir modulación en amplitud o en frecuencia (FM o AM), e incluso, simultáneamente los dos tipos de modulación. Actualmente existen ampliaciones de las capacidades de sonido, ofreciendo calidad de 16 bits en sampleo (calidad CD) y 16 canales, así como interfaz MIDI.

- Controlador (en chip Paula) de unidad de disco vía DMA basado en la lectura o escritura de pistas completas. Esto significa que se pueden leer hasta 5600 bytes en una revolución simple del disco (11 sectores de 512 bytes cada uno). Este controlador actúa de forma independiente al microprocesador M680x0, liberándole de la tarea relativamente lenta de acceder al disco.

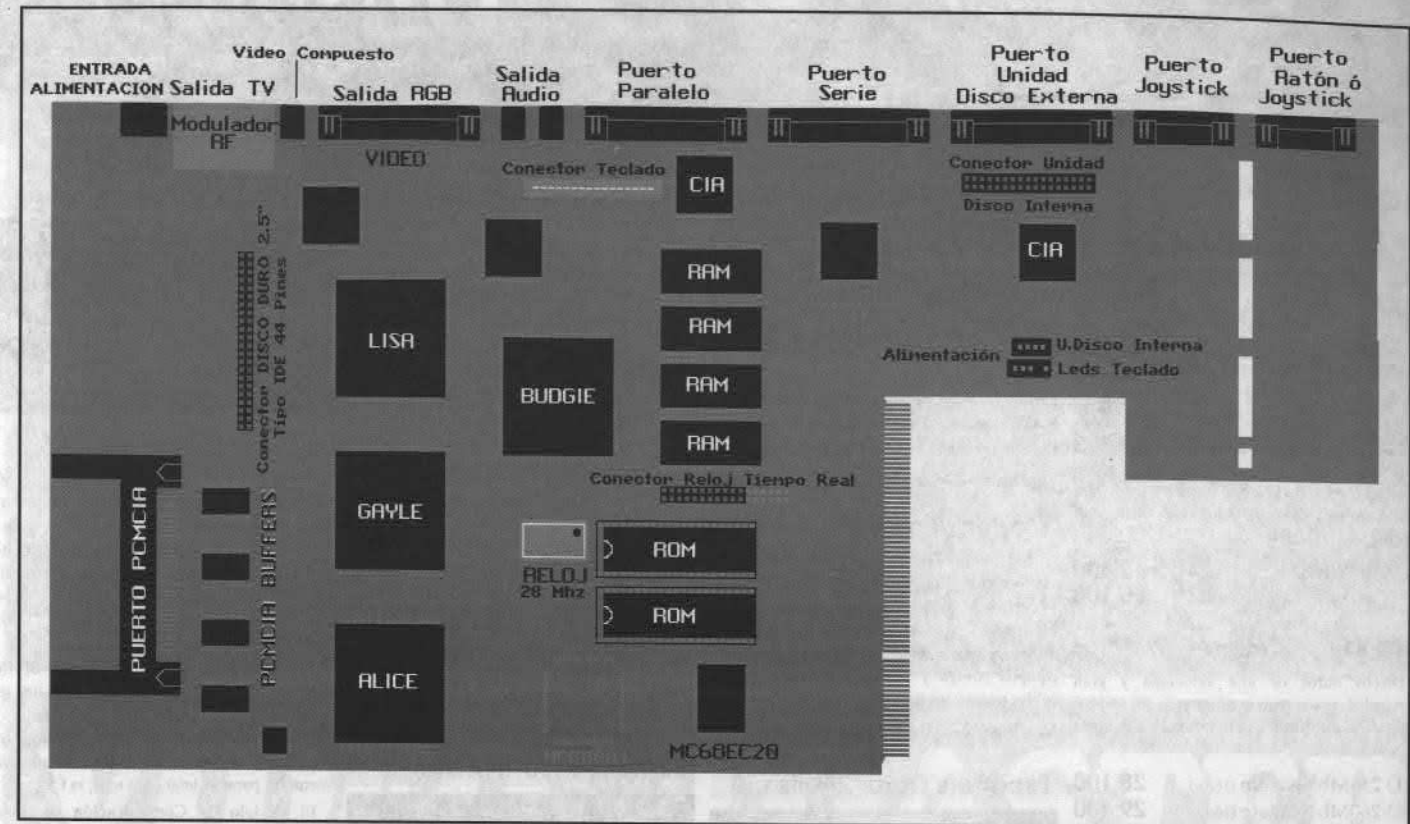


FIGURA 3. Si tomamos como ejemplo de integración en placa los últimos modelos de Amiga, como el A1200/A4000, observaremos que es muy elevada, puesto que los principales circuitos de todo el ordenador residen en media docena de pastillas (chips) de silicio. En este esquema (a escala) de la placa base del A1200 los chips cuadrados (custom chips) denominados "lisa, gayle, alice..." contienen en su interior cientos de miles de transistores. Como anécdota, comparar el tamaño del chip MC68EC20 frente a cualquier custom chip... más de uno de nosotros se habrá llevado una gran sorpresa al comprobar que el microprocesador de 32 bits pasa casi desapercibido entre los "mastodontes" como "alice" o "lisa". También hay que comentar que todos los chips están soldados directamente en la superficie de la placa (sin zócalo, excepto las dos ROMs del S. Operativo) y por lo tanto aquellas reparaciones caseras o montajes que podíamos realizar en un A500/A2000/A3000 (¡como cambiar los martirizados CIAs!) son imposibles actualmente en un A1200/A4000. Requieren soldadores especiales y sobre todo muy caros.

HABLEMOS DE TECNOLOGÍA...

Si observamos a simple vista el interior de cualquier Amiga, la primera impresión que se puede tener es la de curiosidad al comprobar que la placa madre, lejos de estar repleta de componentes electrónicos, está casi vacía y recorrida por infinidad de pistas. Sin embargo, hay una docena de pastillas de silicio negras, rectangulares o cuadradas, que esconden en su interior el motor del Amiga. En cada una de estas pastillas, circuitos integrados o 'chips', residen principalmente transistores de tamaño microscópico, que procesan a nivel de bit toda la información que entra y sale.

Este sería el nivel más bajo al que podemos llegar: el nivel de transistores. Pero este nivel es demasiado bajo para nosotros, y lo dejamos a los diseñadores de circuitos integrados de Commodore... Las funciones, la programación y las interconexiones entre cada uno de estos chips será nuestro nivel de trabajo, aunque de vez en cuando bajaremos o subiremos de nivel para explicar el porqué de algunos detalles.

La moderna tecnología empleada para fabricar estos chips permite que el nivel de integración en una base de silicio de 1 cm² sea muy alto, del orden de 100.000 a 1.000.000 transistores, situándose en la escala VLSI (o Very Large Scale Integration), utilizada actualmente por la mayoría de fabricantes de ordenadores. Gracias a esta tecnología, es posible compartir y aprovechar el espacio de una pastilla para colocar circuitos con funciones totalmente diferentes, por ejemplo, el chip de Amiga Agnus se

compone de varios circuitos distintos (¡ya veremos más adelante para que sirven!). ¡Ojo! Cuando observamos los chips del Amiga vemos que algunos son considerablemente grandes, por ejemplo, algunos de hasta 4cmx4cm y 8cmx3cm, pero todo el corazón del chip reside en un área de silicio de aproximadamente 1 cm², el resto de superficie es empleada para poder colocar en ella las 'patitas' o pines de conexión al exterior (mientras más pines tenga un chip, mayor será su tamaño...aunque también juega un papel importante la cantidad de calor que el chip tiene que disipar mientras está en funcionamiento).

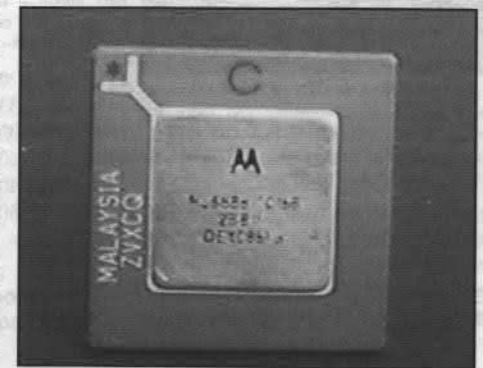
A medida que Commodore ha ido lanzando nuevos modelos de Amiga, el diseño y fabricación de las placas ha ido evolucionando: primero usó la típica placa con los componentes en una cara y soldados por la opuesta, pistas en las dos, y componentes colocados en zócalos... (por ejemplo los A500 o A2000), y actualmente utiliza la moderna técnica SM (Surface Mounted), donde los componentes ya no residen en zócalos, sino que están soldados directamente en la superficie de la placa, sin atravesarla (por ejemplo en el A600, A1200 y A4000). Esta última técnica es mucho más fiable en todos los aspectos y principalmente permite economizar espacio ocupado, hecho evidente cuando vemos que el tamaño de la placa de un A500 es ¡el doble que la de un A1200!

El hecho de disminuir espacio de placa a medida que se va integrando con escalas más elevadas y tecnologías de fabricación nuevas, permitirá, en un futuro no muy lejano, incorporar en

el Amiga más "custom chips" que a buen seguro incrementarán todas las excelentes capacidades gráficas y sonoras, así como potencia de cálculo que actualmente posee. Todos tenemos en mente palabras como DSP, AAA, RISC...y más de uno puede que no sepa exactamente su significado, pero sí que sabe que el futuro apunta en esa dirección.

Para finalizar, adelanto que en los próximos números de Amiga.InFo comentaremos de forma particular y en profundidad cada uno de los componentes "hardware" que integran los Amiga.

El primero de ellos será "el microprocesador" y más concretamente la familia Motorola 68000/10/20/30/40. Así pues, en el siguiente número de AI discutiremos sobre partes que componen un procesador, diferencias entre las versiones, e incluso hablaremos de los coprocesadores matemáticos M68881 y M68882... ●



vb soft s.l.
c/ Provenza, 435, 2-3 • 08025 Barcelona

HD IDE 2.5" (A1200)



Discos duros de gran velocidad. Están formateados, e incluyen el cable y el software necesario para crear particiones. Si nos trae el equipo se lo instalaremos.

HD 127Mb Seagate (16ms) **42.800**
HD 209Mb Seagate (16ms) **55.100**

HD IDE 3.5" (A4000)

Discos duros de alta velocidad y gran capacidad, aptos para configurarse como 1ª o 2º disco duro. Ideales para instalar en equipos de sobremesa.

HD 214Mb Seagate (16ms) **28.100**
HD 260Mb Seagate (16ms) **29.400**
HD 341Mb Seagate (14ms) **34.200**
HD 428Mb Seagate (13ms) **35.600**
HD 542Mb Seagate (12ms) **46.500**

HD Fast-SCSI-2 3.5"

Discos duros de altísima velocidad y alta capacidad. Se precisa controladora SCSI-2 ó preferiblemente Fast-SCSI-2 (léase FastLane Z3 o Kit SCSI-2 Blizzard).

HD 545Mb Seagate (12ms) **60.800**
HD 1050Mb Seagate (9ms) **114.800**
HD 1700Mb Seagate (9ms) **214.500**
HD 2100Mb Seagate (9ms) **233.700**
HD 1689Mb Barracuda (8ms) **247.300**
HD 2139Mb Barracuda (8ms) **265.100**
HD 4294Mb Barracuda (8ms) **498.700**

HD Removable SCSI-2

Disco duro removible interno, en formato 3.5", de gran velocidad. Se precisa controladora SCSI-2.

HD 105Mb SyQUEST (14ms) **48.500**
Cartucho 105Mb **11.900**

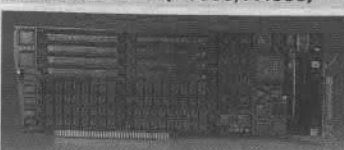
Unidad CD SCSI-2



Unidad lectora de CDs de doble velocidad para Amiga. Es compatible con todos los formatos de CD (también PhotoCD y HFS). Incluye software de control: AmiCDROM. Se precisa controladora SCSI-2.

Lector CD-ROM Toshiba (int.) **37.900**
Lector CD-ROM Toshiba (ext.) **51.200**

FastLane Z3 (A3000/A4000)



Controladora Fast-SCSI-2 (10Mb/s) por DMA. También es ampliable hasta 256Mb de RAM. Soporta todos los dispositivos SCSI-2 (también lectoras CD-ROM).

FastLane Z3 **69.500**

Modem/Fax BZT

Modem/Fax externo v42bis (57.600 bps y 100% comp. AT). Soporta también: v21, v22, v22bis, v23 (ibertex), v32, v32bis, v42 y MNP 1-5; y en fax: Class 1 y 2, Grupo III.

Incluye software para datos y fax, para equipos Amiga y PC bajo Windows, y los cables necesarios para su conexión.

Modem/Fax BZT 14400 **37.900**

Escáners Color 24bits



Escáners color (24bits, 256 tonos y line-art) DIN-A4 de sobremesa. Resoluciones hasta: 1200 y 1600ppp. Diferentes tipos y modos de corrección, brillo, contraste y dithering.

Opción para alimentador automático de documentos y adaptador de transparencias.

Epson GT-6500 (+cable Amiga) **119.900**
Epson GT-8000 (+cable Amiga) **198.900**
Alimentador documentos **76.900**
Adaptador transparencias **112.900**

Memorias (varios equipos)

SIMM 1Mb (1x9) **8.500**
SIMM 4Mb (4x9) **33.500**
SIMM 1Mb (32bits 72pines) **8.000**
SIMM 4Mb (32bits 72pines) **25.900**
SIMM 8Mb (32bits 72pines) **57.500**
SIMM 16Mb (32bits 72pines) **109.900**
SIMM 32Mb (32bits 72pines) **consultar**

Coprocesadores

Coprocesadores matemáticos Motorola, para añadir a las placas aceleradoras o de memoria con esta opción y también a los Amiga 4000. No se incluye el oscilador.

FPU 68882-20/PLCC **12.900**
FPU 68882-33/PGA **14.900**
FPU 68882-33/PLCC **14.900**
FPU 68882-40/PGA **23.500**
FPU 68882-40/PLCC **23.500**
FPU 68882-50/PGA **29.000**
Oscilador (25, 33, 40, 50MHz) **1.000**

Blizzard 1220 (A1200)

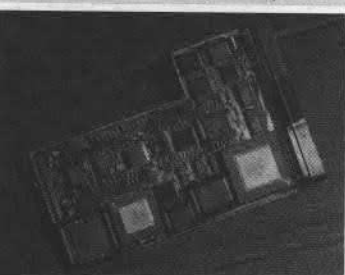


Ampliación de memoria y aceleradora en una sola tarjeta para el A1200.

Incluye 4Mb de 32bits y es ampliable hasta 8Mb con un módulo adicional. También incluye un procesador MC68EC020 a 28MHz (sin MMU) y acepta cualquier FPU 68882 en formato PLCC.

B1220 (4Mb, sin FPU) **49.900**
B1220 (4Mb, FPU 33MHz) **64.800**
B1220/ADD4 (módulo de 4Mb) **34.900**

Blizzard 1230-II (A1200)



Aceleradora 68030 para Amiga 1200 disponible en versiones EC030 a 40MHz y 030 con MMU a 50MHz. Posee dos puertos de expansión: uno DMA para el Kit Fast-SCSI-2 (10 Mbyte/s) y el otro para futuras opciones.

Es ampliable hasta 64 Mbytes de 32bits a través de SIMMs de 72pines, y acepta FPUs de ambos tipos (PLCC y PGA).

B1230-II/40 (sin FPU) **52.900**
B1230-II/40 (FPU 33MHz) **67.800**
B1230-II/40 (FPU 50MHz) **75.800**
B1230-II/50 (sin FPU) **67.900**
B1230-II/50 (FPU 33MHz) **82.800**
B1230-II/50 (FPU 50MHz) **90.800**
B1230-II Kit Fast-SCSI-2 **21.900**

Blizzard 4030 (A3000/A4000)

Aceleradora basada en un MC68030 a 50MHz con MMU y con opción de FPU 68882/PGA también a 50MHz.

Se instala fácilmente en el A3000 ó el A4000/030; sólo hay que insertar la placa en el slot de CPU de estos equipos.

Los procesos y los accesos a la memoria de la placa base son dos veces más rápidos, que los conseguidos por el propio A4000/030.

Utiliza siempre la memoria base del equipo, y es por ello que no requiere memorias especiales o adicionales para poder funcionar correctamente (como otras aceleradoras).

B4030/50 (sin FPU) **59.500**
B4030/50 (FPU 50MHz) **82.400**

CyberStorm (A3000/A4000)



La única aceleradora completamente modular para todos los A4000.

Admite módulos de: CPU (68040 y 68060), Memoria (hasta 128Mb), Caché Secundaria y Comunicación.

Existen módulos de CPU que admiten diferentes tipos de procesadores y diferentes frecuencias (hasta 80MHz). Están en desarrollo módulos con CPU 68060 a 66 y 80MHz, e incluso con RISC PowerPC.

El módulo de Memoria admite SIMMs estándar de 32bits y 72pines, con los que amplía el sistema hasta 128Mbytes.

Gracias al módulo de Caché Secundaria (opcional) se puede expandir esta preciada memoria para acelerar, aun más, la CPU.

El módulo de Comunicación (opcional) incluye: Fast-SCSI-2 (10Mb/s), Ethernet (BNC y AUI) y Serie de alta velocidad.

Es completamente compatible con todo el software y hardware creado para el A4000, y además deja libre todos los slots Zorro-III.

C040/00 (sin CPU 68040) **109.500**
C040/40 (CPU 68040/40MHz) **199.500**
C060/50 (CPU 68060/50MHz) **289.500**

Emplant (A3000/A4000)

Emuladora Macintosh, la cual incluso supera las prestaciones del original. Soporta color, sonido estéreo, puertos AppleTalk, discos duros, tarjetas de 24 bits, periféricos SCSI-2... (según el modelo).

Y además todo en ¡Multitarea Real!!.

Emplant Basic **59.900**
Emplant opción A (2xRS422) **69.900**
Emplant opción B (SCSI-2) **69.900**
Emplant Deluxe (SCSI/2xRS422) **79.900**
Módulo IBM (emulador PC) **24.500**

VideoLab 1.21 (software)

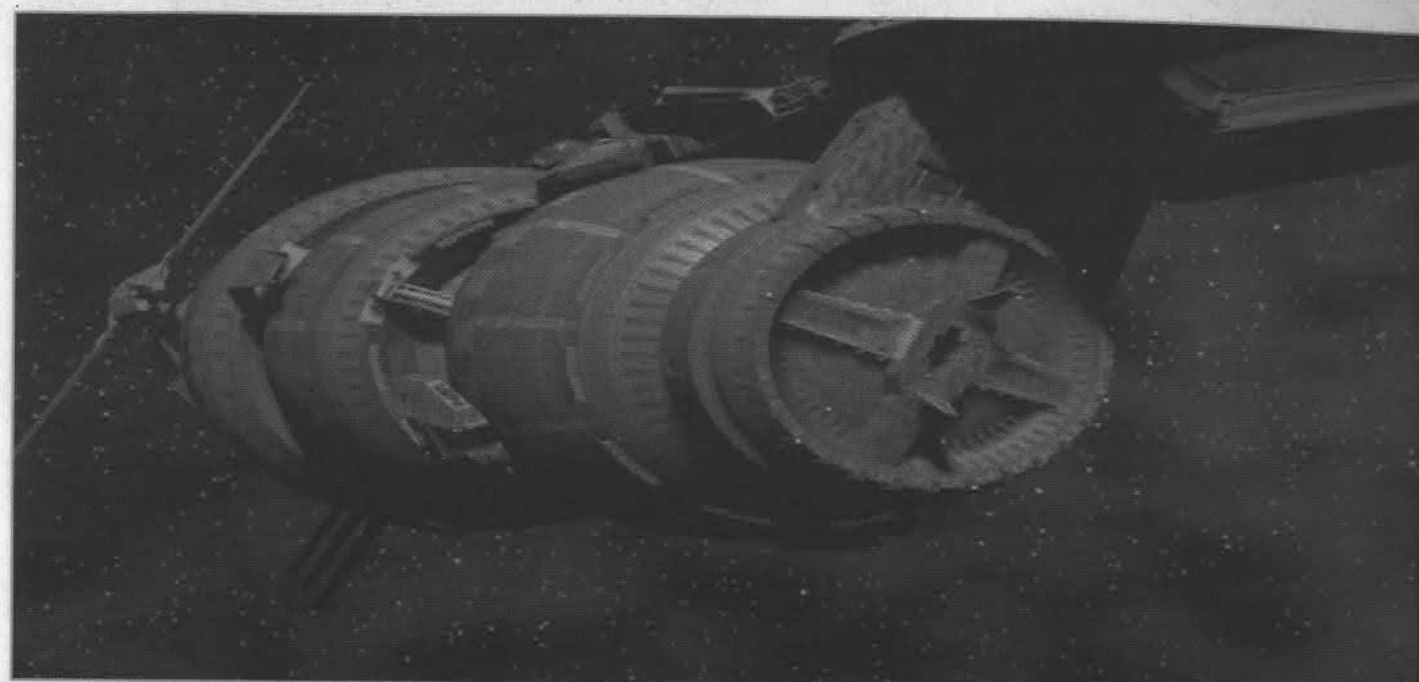
El mejor software de conversión y creación de animaciones para Amiga. Perfecto para utilizar con programas de Multimedia como: AmigaVision, MediaPoint ó Scala.

También le permite grabar sus animaciones a vídeo y en tiempo real.

VideoLab 1.21 (español) **11.900**
VideoLab 1.21 (català) **11.900**
VideoLab 1.21 (english) **11.900**

Títulos CD-ROM (software)

Aminet 2 (Marzo'94) **4.000**
C User's Group Library **6.000**
Corel PCD varios (c/u) **9.900**
Resource Library Audio 1 **4.000**
Resource Library MultiMedia **6.000**



VÍDEO TOASTER



Utilizada por miles de usuarios en EEUU ésta es, sin lugar a dudas, la "tostadora" más potente en creación de gráficos y efectos de vídeo. Fco. Pérez nos habla de de todo lo relacionado con ella.

Cuando en 1990 una computadora que realizaba efectos de vídeo en tiempo real costaba de 50 a 100 millones de pesetas, nadie se podía imaginar, en los Estados Unidos de América, que una nueva compañía formada por Tim Jenison y Paul Montgomery iba a causar una auténtica revolución en la industria de la videoproducción, por menos de 200.000 pesetas y un Amiga 2000.

Todo el mundo centró su atención en NewTek, la empresa bajo cuya protección salió a la luz el proyecto Video Toaster. Nadie escapó de su influencia: desde productores de empresas videográficas, a compañías publicitarias para la televisión o equipos deportivos, centros de educación o productores cinematográficos.

La Video Toaster ofreció 9 herramientas esenciales para realizar una videoproducción de alta calidad: efectos digitales de vídeo en tiempo real, generador de caracteres, sistema de videopintura, gráficos en 3D y animación, dual frame buffers, luma key, procesamiento del color en tiempo real, y digitalización de imágenes en tiempo real.

Digital video Effects (Efectos Digitales en Tiempo Real) con el Switcher

La VT nos ofrece una de sus más atractivas utilidades: la edición digital de vídeo en tiempo real, gracias al Switcher. Consiste en un panel con 9 bancos, con 32 efectos o transiciones cada uno que se pueden aplicar a las dos fuentes de vídeo seleccionadas de cuatro entradas posibles, o entre una fuente de vídeo seleccionada y una fuente de vídeo digital interna del Amiga (Animación del Amiga en Ham8 a 30 frames por segundo).

Como si se tratara de una mesa mezcladora profesional de vídeo, pero con infinitas posibilidades, nos podemos sumergir en un apasionante mundo de efectos increíbles.

Efectos como una rotura de cristales, donde el cristal que se rompe, es la imagen en movimiento que entraba primero y lo que queda detrás, mientras desaparecen los trozos de cristal, es la nueva fuente de vídeo. Tal como pueden ser las roturas de cristales, existen otras transiciones isomorfas como son

el efecto de pantalla ardiendo para dar paso a la siguiente fuente de vídeo o la pantalla que se convierte en líquida y que fluye hasta desaparecer y completar la transición. Ahora, con la versión 3.0 de la VT se ha añadido la posibilidad de acompañar a este tipo de transiciones un sonido que ayude a ilustrar mejor el efecto.

Otros efectos espectaculares son los "page peels" y "flys" que son efectos de rotación, desplazamiento, vuelta de hoja, ondulación, etc. de la fuente presente para dejar entrar la siguiente (por ejemplo, en un noticiario, las noticias pueden entrar una detrás de otra con este tipo de efectos, o ir apareciendo rotadas en tiempo real cada una de ellas por un lado de la pantalla).

Además de estas transiciones clásicas, se han añadido nuevos efectos como la transición con una animación interna: por ejemplo, creamos una animación que acabe con una explosión (un personaje corriendo que acaba accionando un detonador y la pantalla queda iluminada por la explosión). Podemos hacer que cuando seleccionemos esta transición se muestre esta animación y cuando quede la



TEL (93) 456 1545
FAX (93) 456 1545

pantalla en blanco entre la segunda fuente.

Casi 300 efectos espectaculares que harían deshacerse de envidia a cualquier productor televisivo europeo que haya invertido 50 (o más) millones de pesetas en un equipo que ni siquiera hace más cosas que rotar dos fuentes de vídeo en tiempo real.

CHARACTER GENERATOR (Generador de Caracteres)

Es la utilidad de la Video Toaster que permite crear todo tipo de rotulaciones, scrolls, y de efectos tal y como se haría en cualquier editor de titulación (VGM, TV-Text, etc.).

Tras la aparición de Video Toaster 4000 los usuarios tienen a su disposición un sinfín de novedades dentro de esta herramienta: Por ejemplo, existe la posibilidad de incorporar brushes en la generación de caracteres, desde 1x1 hasta 752x400 pixels con el número de colores de la máquina bajo la que se trabaja (con un Amiga 4000 262,144 colores de una paleta de 16 millones). Además, con la nueva VT 4000 y el software 3.0, ya es posible realizar "alpha channel" entre las señales de entrada y lo que hemos generado en el CG (Por ejemplo: Si estamos entrando por dos de las fuentes de vídeo los titulares del día en nuestro noticiario y queremos que la sinopsis de cada una de las noticias aparezca transparente sobre las fuentes de vídeo).

Otra gran novedad, es la posibilidad de utilizar como fuentes las PostScript type 1 y type 3 de Adobe y las Agfa Compugraphic Scalable Outline, además de su propio formato en bitmap. Esto supone usar fuentes entre 10 y 400 líneas de alto, que se pueden escalar, retocar el color y la profundidad de los contornos y el interior de cada una de los caracteres por separados. Ahora usar el CG es tan fácil como escoger un tipo de fuente y ponerse a escribir cualquier cosa y ya está.

Toaster Paint (Sistema de VideoPintura)

Es un editor de imágenes de 24 bits en el que se puede hacer todo tipo de efectos y manipulaciones sobre gráficos en 24 bits para después utilizarlos como brushes o como imágenes estáticas para insertar sobre las fuentes de vídeo, con la posibilidad de, entre muchas otras opciones, incluir máscaras (bits) de transparencia en nuestros trabajos para usar con el Alpha Channel.

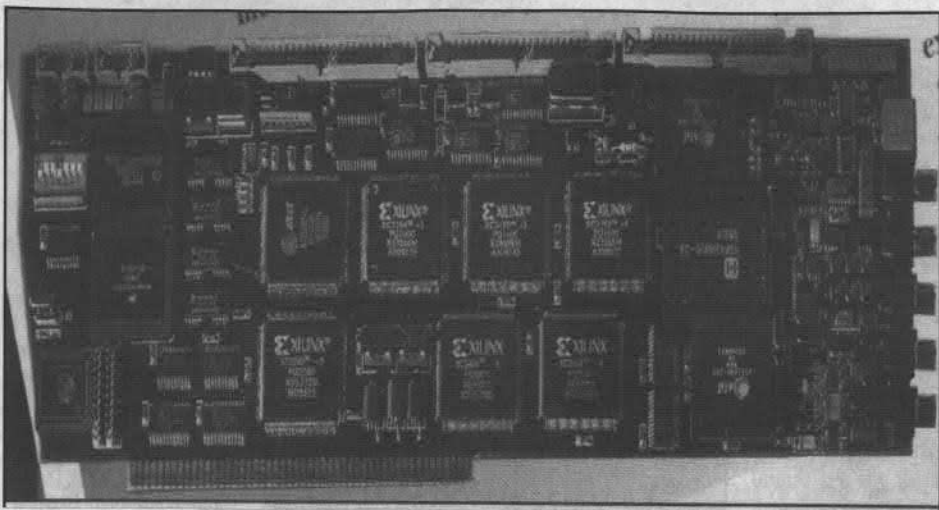
Es facilísimo usar el frame grabber para congelar la imagen que nos entra por una de las entradas de vídeo e inmediatamente retocarla para usarla en el mismo instante.

Lightwave 3D (Gráficos en 3D y Animación)

Gracias a un hardware especial interno de la VT, el sistema de generación de gráficos en 3D y animación Lightwave, desde el primer momento, supuso una auténtica revolución en el mundo de las animaciones realizadas por ordenador.

En los EEUU, la tecnología en el vídeo ha llegado a unos grados de sofisticación en los

Realmente ha de ser complicado el diseño de una tarjeta con tantas prestaciones. La mayoría de los "chips" especiales que la componen son exclusivos de NewTek. Únicamente está disponible de forma interna y versión NTSC para A2000, A3000 y A4000.



que es más fácil que un joven coja del videoclub una película llena de efectos especiales hechos por ordenador, y que se sienta altamente atraído por ella, antes que querer ver una serie de animación clásica (con maquetas a escala). A los productores les saldrá el espeluznante coste de una maqueta de una nave espacial a escala por el mismo precio que un sistema VideoToaster completo.

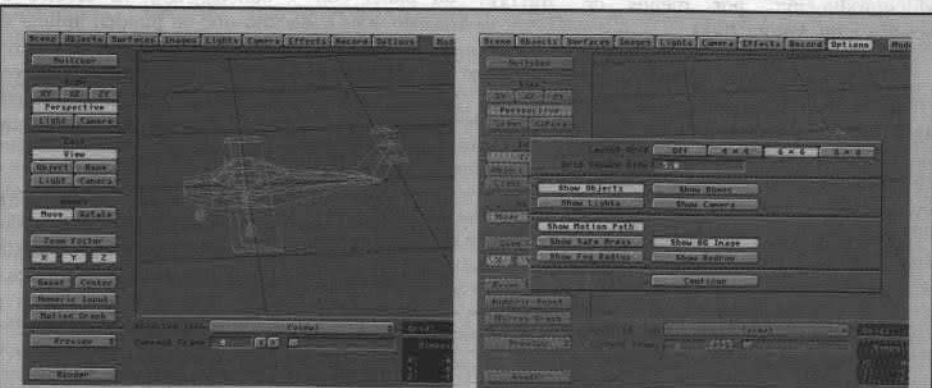
De hecho, series en las que sólo se utilizaba el Amiga/VT para hacer efectos de segunda categoría, como inserción de estrellas de fondo moviéndose a gran velocidad en la sala de control de 'Star Trek the Next Generation', después de hundirse gracias a los precios abusivos que exigía la gente de Silicon Graphics (1 millón de dólares por animar unas cuantas veces la Enterprise en cada episodio), han reaparecido con efectos íntegramente realizados con la VideoToaster ('Next Generation' reaparece como 'Deep Space 9', y ¡por solo 100.000 dólares por episodio!).

En el capítulo piloto del proyecto Babylon 5 (serie que este 1994 ya ha sido una realidad en USA, y está rompiendo esquemas en la concepción de la Ciencia Ficción: os recomiendo el capítulo 8 'And the Sky Full of Stars'), Foundation Imaging ya nos impresionó con las escenas alucinantes que se desprendían

de la serie: renders increíbles que parecen mentira que unos modestos usuarios de Amiga hayan realizado. La imagen de la página uno es una muestra de una de las naves., Ha sido extraída del vídeo que ya podemos obtener en los videoclubs nacionales y en castellano.

Steven Spielberg tampoco se ha escapado de esta ola de gran calidad a precios mínimos. Después de que Amblin Entertainment encontrara en 'Parque Jurásico' un negocio gracias al modelado con Amiga e Imagine (barato y competitivo) y el render de los dinosaurios en Workstations Silicon Graphics (Bastante caro, pero hasta la fecha sin poderse realizar de otra forma por motivos temporales), el nuevo proyecto fue una coproducción entre Amblin y Universal Television, llamado SeaQuest DSV (Deep Submergence Vehicle), una historia de investigación submarina de civilizaciones perdidas sumergidas en el fondo de los océanos.

En este proyecto, se pudo utilizar íntegramente el ordenador Amiga gracias a lo que se llamó "the render farm" (la granja del render), una sala no muy grande donde conviven 40 VideoToasters con procesador 68040 a 33 Mhz. cada una, controladas con el software RenderMaster de Nutopia para crear prioritariamente las animaciones que se



Lightwave. El editor de objetos es una de los más intuitivas y potentes características que esta pieza de arte en programación ofrece al usuario. Una maravilla que demuestra hasta dónde puede llegar el Amiga si se toma seriamente.

Video Flyer 4000: La edición no lineal en VT

El proyecto Video Toaster, por muy espectacular que pareciera, no fue nada en comparación con lo que podía haber sido de no ser obstaculizado por el nivel de desarrollo tecnológico que había en 1991: ¿Quién no recuerda aún lo desorbitado de los precios de discos duros? ¡lo que se tenía que llegar a trabajar para conseguir 40 miserables Megs!

Paul Montgomery y Tim Jennison no escatimaron imaginación a la hora de crear su gran invento: Habían previsto que la VT se conectara con un sistema de edición no lineal de vídeo, integrado completamente en el sistema, y tan fácil de utilizar como la VT. Desgraciadamente, gracias a estos desavances tecnológicos, no se pudo acabar bien el trabajo, y la VT salió tal y como la conocemos.

Pero ahora, en 1994, las cosas han cambiado bastante. Comenzamos a ver en el mercado sistemas de almacenamiento masivo que pasan del giga, y que llegan a valer como lo que en nuestros comienzos encontrábamos suficiente tanto en cantidad como coste. El nuevo producto se llama Video Toaster Flyer. Es un editor de vídeo no lineal, con calidad de vídeo digital D2. Sin llegar a valer los \$4000, tenemos la solución a toda la complejidad que supone un sistema de edición lineal (A/B Roll).

Es una tarjeta compresora en tiempo real, que integra 3 interfaces SCSI para 21 unidades de almacenamiento de datos, que interacciona con la VT con una completa integración gracias a que, ya desde un principio, se diseñó el sistema para soportar edición de vídeo no lineal. Las entradas de vídeo aceptan vídeo compuesto y S-vídeo, y tiene dos salidas que se conectan directamente a dos de las cuatro entradas de la VT, en una completa interacción. El método de compresión se llama Video Toaster Adaptive Statical Coding (VTASC). Ha sido desarrollado por NewTek, y no se basa en las transformadas del coseno como los demás existentes, manteniendo calidad Broadcast. El efecto de aumentar el factor de compresión es de una suavización de la definición de la imagen, sin aparecer extraños efectos como en los otros.

El modo de funcionamiento consiste en que se entra en la Flyer el Audio y el Vídeo con la base de tiempo corregida. El vídeo se digitaliza y comprime, y el audio se samplea y almacena en otra parte sin comprimir. Se dispone de bancos como los de los efectos de transiciones del switcher, con uno de los frames de la animación.

En uno de los 4 modos de funcionamiento, para preparar un script con los distintos fragmentos digitalizados, se mueven los "iconos" de los clips hacia la zona del proyecto, y se les asigna un efecto o transición como se hace normalmente con la VT.

SOFTWARE VERSIÓN 3.5

Gracias a la nueva versión 3.5 de LightWave tenemos todas estas nuevas utilidades añadidas:

- ☐ Tiempos de renderizado de hasta 10 veces más rápido.
- ☐ El uso de huesos (bones) para crear sistemas esqueléticos.
- ☐ Lens flares con glows ajustables, streaks, reflexiones y anillos.
- ☐ Antialiasing ajustable rápidamente. Morphing de superficies. Rendering de campo.
- ☐ Motion Blur con shutter time ajustable. Profundidad de Campo con distancia focal ajustable.
- ☐ Motion paths visible en el Layout. Grabación con alpha channel y composición de exterior alpha.
- ☐ Posibilidad de aplicar mapas de deformación para cambiar objetos.*
- ☐ Nuevo interfaz con el usuario más fácil de usar con menús desplegables.
- ☐ Se comparten objetos directamente con el modeler.
- ☐ Control interactivo con ratón de tensión de splines, continuidad y bias.
- ☐ Cono de vista spotlight en layout. Es posible cambiar la vista para lo seleccionado por la cámara.
- ☐ Test de render de una región limitada. Modo sketch interactivo.
- ☐ Trace image backgrounds en el modeler. Soporte del texto en postscript.
- ☐ Funciones booleanas, suma, resta, unión e intersección. Herramientas de taladro y taladro sólido.
- ☐ Hacer un extrude a un path solo. Modelaje con Spline path. Alineamiento de polígonos automático. Herramienta de Smooth shift.
- ☐ Nuevo interfaz rápido e intuitivo. Herramienta de array, macros arexx desde los menús y muchas novedades más.



necesiten. Las animaciones se almacenan temporalmente en 10 discos de 1.2 Gigas para después volcarlos a un CDVideo grabable desde el cual ya se pasa directamente a la post-producción. En el momento de la redacción del artículo, ya se había trasladado un prototipo del casi utópico "screamer" (aceleradora que incorpora procesadores en paralelo) para que se genere con mayor velocidad cada capítulo semanal (canal 5, NBC, 7 p.m.).

Cuando entras en el mundo de LightWave, por poco experto que seas, verás que cualquier persona que se acerque a este tipo de aplicaciones puede realizar los trabajos

que quiera, por difíciles que le parezcan, y aunque no tenga ninguna noción de 3D, informática, hardware, y sólo sepa lo que normalmente realiza, la videoproducción.

Esa es la realidad. Hay aplicaciones que son muy difíciles de usar, pero te hacen todo perfectamente bien. Tienes control sobre el mínimo parámetro, pero una persona que viene del mundo del vídeo no sabría utilizarlas porque no le interesan las cosas minuciosas. Quiere hacer una animación de un logo volando a través de un escenario, y que de repente vaya contra el espectador de forma rápida, por ejemplo, y lo quiere realizar de forma fácil, casi automática. Quiere que el

render no sea para morirse esperando. Quiere LightWave.

Dentro de 3D está el Modeler, el módulo que nos permite crear objetos en 3D, aplicarles herramientas, realizar sobre ellos operaciones, etc. como cualquier programa de raytrace que conozcamos. Existe la posibilidad de entrar directamente caracteres por teclado y operar con ellos en forma de objetos en 3D, trabajar con los Bones (huesos), que es una de las muchas características que se han añadido a la VT en su tercera versión. Consiste en definir los huesos que componen un objeto para luego animarlo de forma independiente a su estructura, únicamente teniendo en cuenta las articulaciones que forman los huesos, como si de un esqueleto se tratara. De esta forma, podemos hacer que cualquier objeto no animable normalmente (como las latas de bebida Shasta Diet en un anuncio de la televisión de USA), pueda moverse con una naturalidad alucinante.

Aparte de los bones, están los "Lens Flare", que son manipulaciones sobre los atributos de las lentes de la cámara. Fácilmente podemos modificar los parámetros de la cámara para poder desenfocar, crear los posibles reflejos internos de la óptica de la cámara, modificar luces, estrellas, soles, etc. Además se pueden realizar todo tipo de efectos sobre los objetos en 3D, como morph, warp,

¿ QUE SIGNIFICA ...?

✓ **Alpha Channel** es un método empleado para aprovechar los 8 bits que sobran en la palabra de 32 bits de las estaciones de trabajo con gráficos de 24 bits. Estos 8 bits se utilizan para crear el nuevo atributo que es la transparencia lineal o "alpha channel". Sirve para poder hacer mezclas con una fuente de vídeo externa u otras imágenes, y que el nivel de transparencia u opacidad quede determinado por la combinación de estos bits. Puede ser de todos los bits que se necesiten (la aplicación Alpha Paint utiliza 36 bits por pixel, con 12 bits de alpha channel).

✓ **Luma/Croma key** es una técnica donde una de las señales de entrada de vídeo se mezcla con otra creándose una ventana por la cual pasa la señal de otra entrada. En Luma Key, se deja pasar la señal por los píxeles que cumplen la propiedad de una luminosidad concreta y en el Croma key el método es mediante un color concreto.

✓ **NAB**, del inglés National Association of Broadcasters, es un fórum anual que realiza la asociación americana de vídeo productores, donde se presentan novedades y productos.

✓ **Render** es el acto de generar una imagen o animación. Normalmente se utiliza el concepto de "Tiempo de Render" para tener una medida del tiempo empleado en generar esa imagen o animación.

✓ **TBC** del inglés Time Base Corrector, corrector de base de tiempo. Es un dispositivo electrónico que se utiliza siempre que se trate con señales de vídeo. Su función es la de intentar evitar desajustes de la señal de vídeo que le entramos, y corregirlos, tanto en color como luz, así como evitar que deje de entrar señal al dispositivo al que tenemos conectada la salida del TBC.

El passport 4000

Por fin llegó la auténtica compatibilidad entre los sistemas NTSC, PAL y SECAM. Gracias a la compañía americana Prime Image Inc. podemos disfrutar la Video Toaster en Europa. Su producto, Passport 4000, es un multiconversor completamente digital de señales, que incluye TBC's y sincronización. La estructura externa de la unidad es una caja tipo sobremesa que monta un panel frontal-botonera de control. Es independiente y tiene fuente de alimentación propia.

En su interior, alberga, además de todos los componentes, slots de expansión tipo zorro-III como los del Amiga 4000, para poder insertar otras tarjetas, como si de una expansión de los slots se tratara. Los tipos de señales de entradas y salidas con los que puede trabajar son PAL, PAL-M, PAL-N, SECAM NTSC y NTSC 4.43.

La Passport 4000 opera con más de 6 campos de memoria por canal, usando la técnica propia de interpolación de Prime Image "pass through", lo que asegura que la señal se transforme íntegramente a lo largo del proceso, sin ningún degradado como a los que estábamos acostumbrados a notar con los conversores clásicos.

El precio, en un principio, parece abusivo (\$6700, más o menos un millón de pesetas), pero teniendo en cuenta que incluye TBC's en cada una de las entradas, y que puede actuar además, de frame grabber, se puede considerar bastante asequible para realizar producciones con VT, gran calidad, y compatibilidad con todo el planeta.

etc. Por último, cabe destacar el "Motion Blur", la posibilidad de animar grupos de objetos dándoles cualidades físicas como peso, gravedad en el ambiente, viento, etc. especial para copos de nieve, estrellas, hojas secas. En definitiva, todo lo que queramos controlar directamente pero que hemos de automatizar para liberar el trabajo.

Como complemento al Motion Blur, han aparecido en el mercado (americano) otros paquetes de software que trabajan más perfectamente con las leyes físicas de la materia, y que dan una apariencia más real al movimiento que realizan los objetos.

Dual Frame Buffering

Mediante el hardware implementado con dos buffers, la VT puede mantener dos imágenes de 24 bits en alta resolución en pantalla. Con esta aplicación podemos permitir utilizar estas imágenes como máscaras que dejarán pasar las fuentes de vídeo, para aumentar así las posibilidades en tiempo real del sistema.

Luma Key

Se pueden realizar luma keys con fades incluidos, secuencias animadas bajo luma y muchos efectos más que escapan de la típica imagen del hombre del tiempo sobre un mapa. El croma key (por color) es posible mediante la adquisición de módulos externos de terceros fabricantes.

Color Processing

Se pueden alterar en tiempo real los atributos del vídeo: color, brillo, contraste, negativizar, solarizar, croma, de día a noche, etc., un gran número de efectos que complementan el abanico de posibilidades de la edición de vídeo en tiempo real. Posibilidades que han permitido la realización de 'vídeos' musicales con infinitos efectos en tiempo real, que son los que realmente atraen al público.

AUSTRALIA VERSUS USA

La compañía australiana Opal technology, Ltd., simultáneamente al éxito de la Video Toaster, creó una conocida tarjeta de vídeo de 24 bits, llamada Opalvision, que posteriormente fabricó y distribuyó la compañía californiana Centaur Development.

No sería esta noticia singular de no ser por el anuncio, ahora hecho realidad, de la aparición de unos módulos de expansión que se conectarán a la tarjeta en forma de módulos internos adicionales, que realizarán un servicio parecido al de la VT.

El Video Processor es una placa con apariencia externa muy similar a la Video Toaster 4000 (con hueco para que quepan los SIMMs de memoria en el A4000). Es un procesador de vídeo que se conecta en el slot de vídeo del Amiga (la Opalvision no se instala allí, sino en cualquier slot zorro libre). Este módulo realiza las tareas de genlock de calidad broadcast, luma y croma key, Video Sandwich key, alpha channel de 8 bits, efectos de procesamiento de color en tiempo real, número ilimitado de transiciones y efectos digitales en 3D, picture-in-picture de 24 bits, y caracter generador de 1ns, 24 bits. Incluye un chip llamado Roaster, que es el corazón del sistema. Se encarga de crear los efectos tiempo real, y de realizar cientos de tipos de transiciones, entre las que están cortes, fades, efectos orgánicos especiales, borrados, page peels, etc.

La Opalvision Video Suit, es un módulo de tipo rack montable en un estudio, que se conecta directamente al Video Processor. Incluye un mezclador de vídeo y audio con hasta 9 entradas de vídeo y 10 de audio, y varias salidas de vídeo y audio. Todas las entradas y salidas se pueden configurar como RGB, luminancia y crominancia, vídeo compuesto y S-Video. Contiene los circuitos que permiten la congelación de frames en tiempo real, los intercambios entre señales (switching), y parte de las transiciones.

Realmente, es la competencia más directa que existe hacia el caballo de batalla de Newtek, la apariencia que ofrece es extraordinaria, y lo que es más importante para los lectores potenciales (esta publicación se lee "mayoritariamente" en Europa): es compatible con el sistema PAL de vídeo, aparte de NTSC, lo que significa una total compatibilidad con nuestro sistema sin necesidad de complicados y carísimos sistemas de conversión (ver columna sobre Passport 4000). Este hecho, además, puede beneficiar a los amantes de la VT. Con la competencia y la introducción de esta tecnología en el Viejo Continente, Newtek ya estará planteándose la colonización de Europa antes de que Australia gane la batalla contra U.S.A. en territorio Europeo en el campo de los efectos digitales de vídeo. De hecho, el actual presidente de Newtek, Tim Jenison ya ha anunciado que la aparición de Flyer 4000 no es sino el inicio de una gran tormenta de nuevos productos, que incluyen pensamientos hacia Europa, y que van a revolucionar la vídeo producción.

Still Store/Frame Grabber

Se pueden digitalizar en tiempo real imágenes en alta resolución y 24 bits para almacenarlas o utilizarlas inmediatamente en el switcher con los efectos digitales de vídeo. Podemos digitalizar un frame de la animación de una noticia de la que tenemos que informar y usarla inmediatamente en el ToasterPaint para modificarla, y más tarde presentarla retocada y con Alpha Channel presentar todas las transiciones por alguna viñeta transparente que le hayamos practicado.

CONCLUSIONES

Por supuesto todas estas realizaciones, e infinitas más, se pueden ampliar gracias a la posibilidad del AREXX, el sistema de comunicaciones entre programas del ordenador Amiga. Con AREXX se pueden realizar todas las cosas imposibles con la VT, y por ello han aparecido paquetes desarrollados especialmente para aprovecharlo: T-Rexx professional, ScriptGen, ToastMaster, Toaster Toolkit 4000, AmiLink (sistema de edición lineal de vídeo), etc.

Se puede sincronizar sonido en 16 bits con la tarjeta Sunrize AD516 editándolo en tiempo real, utilizar la VT como un sistema de edición no lineal de vídeo con la 'framemachine' en multitarea y controlada con AREXX, utilizar el Bars & Pipes como editor Multimedia con su accesorio Toasty, y un sinfín de opciones más.

Ahora, todo esto nos queda un poco lejos, pero dentro de poco, NewTek necesitará expandir su mercado, ante la competencia del mundo PC y Mac, que cada vez se acercan más a los 15 frames por segundo (un Amiga soporta 50 frames/sec.!) a precios que no sean de locura. Después de convencerse de que en el Viejo Continente hay posibilidades, intentará desarrollar un sistema para la Europa PAL, aparte de los multiconversores de norma que ya ha creado y que no son sino un anuncio de la gran ola que va a venir.●

ALTERNATIVAS Y COMPLEMENTOS...

Una tarjeta que se acerca a la VT es el Video Machine de Fast Electronic US, Inc. Un sistema de edición de vídeo para PC's compatibles y Mac's, con la que se pueden realizar A/B rolls (mezclas de vídeo) titular, mezclar y editar vídeo con bastantes transiciones (no se pueden comparar ni en número ni calidad a las de VT o Opal), de entre 6 entradas de vídeo NTSC o PAL. También trata con la señal de audio a la hora de hacer la mezcla. La solución estaría bien si la máquina sobre la que se monta no fuera un arcaico PC o mal diseñado Mac completamente inferiores tecnológicamente a la propia tarjeta. Esto puede causar algunos problemas indirectos, como la utilización del sistema Windows para gestionar los ficheros, en la versión PC, o del Sistem 7, en el sistema Mac. Además, ninguno de los dos tiene una multitarea potente como la del AmigaDOS.

Otros nuevos productos que han aparecido para este entorno Video Toaster, son:

□ **Crouton Tools** es un sistema operativo para la VT, que hace que los usuarios de este sistema tengan a mano todo tipo de utilidades (más de 60) para utilizarlas en la máquina. De entre ellas, están los 3 programas de manipulación fotográfica ADPro, ImageFX y ImageMaster R/L, entre muchos otros.

□ **Raptor** es un sistema acelerador basado en RISC para re-renderizar animaciones con Lightwave. Incluye 2 procesadores R4400 de MIPS que, según el fabricante, acelera la realización de los trabajos entre unas 10 y 15 veces el tiempo normal. Ofrece Windows NT como sistema operativo, y 1 Mg de memoria cache secundaria y 128 megas de RAM. Se conecta al Amiga por Ethernet, ofreciendo la posibilidad de enlazar 20 Raptors para formar la granja del render más potente que pueda existir, a un precio que no es mucho más que el de una estación Unix de SG.

Esta es la lista de direcciones de las compañías que aparecen en el artículo:

Centaur Development (Opalvision)	P.O. BOX 3959 Torrance, CA 90510 (310)787-4530
NewTek	215 SE 8th St (Video Toaster) Topeka, KS 66603 800-843-8934
Prime Image Inc. (Passport 4000)	19943 Via Escuela Saratoga, CA 95070 (408) 867-6519

ULTIMA HORA: NewTek ha decidido comercializar LightWave como producto independiente de la VT. Comienza así la apertura hacia Europa, puesto que la versión 3.5 de LightWave está disponible para todos en PAL y en España, de la mano de PiXelSoft (Tlf: 979 71 27 00). Ahora, con esta nueva versión, Imagine 3.0 y Real 3D 2.0 van a tener compañía en la carrera por ser el mejor software de creación de imágenes 3D de cualquier plataforma (incluyendo la mítica Silicon Graphics, que está sorprendida por las ideas innovadoras que incorporan estos programas de Amiga y la facilidad de su uso, para obtener resultados de similar calidad).

La parte teórica de la programación



¿Te gustaría saber cómo se programa de forma teórica en la facultad de

Informática? ¿Si? pues estás de suerte, ya que Txinto Vaz nos introduce en este campo bastante desconocido...

Estamos en los 90. Atrás han quedado los 60, los 70 y los 80. La industria informática ha adquirido una cierta madurez y las máquinas de que disponemos han adquirido una potencia considerable. Es evidente que los tiempos han cambiado y que las soluciones de antaño se nos quedan pequeñas. El software, como el resto de componentes del mundo informático, ha evolucionado adaptándose a las nuevas necesidades. Hoy en día es imprescindible para cualquier empresa disponer de un buen soporte informático, y que ese soporte FUNCIONE. No nos hemos de quitar de la cabeza que cualquier utilidad mínimamente seria, sobre todo las orientadas a ayudar a los demás programadores, como compiladores, debe de tener una seguridad de funcionamiento, que en algunos casos es, incluso, más importante que la propia eficiencia del programa en sí. En lenguaje de calle, hemos de intentar por todos los medios que el programa no "pete". Las consecuencias de una mala programación están a la vista de todos: los cuelgues del sistema, los gurus, los bugs de programas, etc. Hay fallos enormes en los programas actuales, ¿quién no se ha encontrado con uno? El usuario pensará que quizá no tiene tanta importancia la necesidad de asegurar el correcto funcionamiento de un programa, pero quizás unos cuantos ejemplos le hagan cambiar de opinión:

Ejemplo n° 1: Usted ha diseñado un programa que calcula una simple división y no tiene en cuenta que el usuario pueda insertar el valor 0 en el denominador. Evidentemente, el programa puede bloquear la máquina en ese caso, debido a que no es capaz de trabajar con números infinitos, y causará un desbordamiento de la variable. Quizá, si se ejecutase en un entorno como MSDOS, el programa no tendría mayores consecuencias: se reinicia la máquina y ya está. Pero... ¿y si el fallo se produce en su Amiga, que dispone de multitarea, cuando está ejecutándose Real 3D o Imagine y se encuentra a punto de terminar ese fotograma que ha tardado 19 horas en crear?.

Ejemplo n° 2: Imagínese que usted construye un programa para llevar la contabilidad de la empresa de un amigo o de un cliente y que, debido a un "bug" del programa, destruya los datos de 9 meses de trabajo. ¿Es importante o no una buena programación? Y llegando a los extremos más insospechados: ¿qué me diría de un "bug" de

similares características en el ordenador que controla la central nuclear más cercana a su casa? En resumidas cuentas, la programación de ordenadores puede llegar a ser asunto de seguridad nacional. Además, hay que remarcar que los programas que sirven de plataforma a otras aplicaciones tienen una necesidad mucho mayor de fiabilidad, ya que los bugs se van propagando de programa en programa, y los fallos ocultos de un sistema operativo o un compilador pueden convertirse en fallos en la ejecución de programas creados para esas plataformas. ¿Se imaginan el programa de la central nuclear ejecutándose sobre Windows?

Diferentes tipos de errores en los programas

✓ **Errores de algorítmica:** Son errores que comete el propio programador a la hora de diseñar el programa. Están causados por la "negligencia" del propio programador. El primer ejemplo del apartado anterior es un claro retrato de este tipo de error. El programador ha de prever los posibles usos del programa y gestionar bien todas las posibles entradas y salidas. Se supone que el usuario va a confiar en los resultados que la máquina le devuelva, con lo cual, en caso de prever que un caso pueda no resolverse satisfactoriamente, el programa ha de comunicar al usuario que esa entrada produce error, o no aceptarla, o comunicar de antemano al usuario que la haga: "El programa actual realiza la división entre numerador y denominador siempre que éste sea diferente de 0", o "Error: No se puede calcular el resultado", o "Error: Ha insertado 0 y no es una entrada correcta",.... Estos artículos están dedicados a evitar este tipo de errores.

✓ **Error tipográfico:** Error que se comete de forma involuntaria al teclear o introducir el programa al ordenador. El principal encargado de detectar estos errores es el compilador, aunque el propio lenguaje puede ayudarnos. Cuando el programador equivoca un símbolo, es normal que el compilador se lo comunique con un mensaje de alerta o error. También tiene que ver el propio lenguaje en ello. Si en vez de una comparación metemos en un condicional una asignación, ésta será más o menos detectable según el propio lenguaje que se esté usando. Esto se tratará en una próxima sección.

✓ **Otros errores:** Pueden contarse entre

ellos desde un corte de electricidad, una bajada de tensión, hasta un mal diseño de una placa del ordenador o del sistema operativo, etc... Este tipo de errores no pueden ser corregidos por el propio programador, pero es bueno que éste intente evaluar algunas de las posibles situaciones, como por ejemplo, las diferentes configuraciones que puede tener el ordenador sobre el que se va a ejecutar el programa, etc.

Elección de un lenguaje de programación

Hay lenguajes para todos los gustos, algunos de ellos principalmente orientados a aplicaciones específicas (como el Cobol, Fortran, etc...) y otros orientados a todo tipo de propósitos (Basic, C, Pascal, Modula-2, etc...) Unos se dejan manejar más que otros y, aunque parezca mentira, los que menos se dejan manejar suelen ser precisamente los más indicados para una programación fiable. Una aplicación escrita en Ensamblador o en C es mucho menos fiable que otra diseñada en Modula-2 o Pascal. También es cierto que, potencialmente, se pueden hacer más cosas en C que en Modula-2, e igual de fiables, pero para ello se necesita mucha más dedicación en lo que a corrección de errores se refiere. Este pequeño ejemplo lo ilustra claramente:

□ En C:
If (!(sinerrores==TRUE)) {salir()}

El programador ha diseñado una función que cuando detecta que la variable booleana 'sinerrores' se encuentra a FALSE, sale de la función, y, por error tipográfico, en vez de '=' (igualdad booleana), ha escrito '=' (asignación). El programa asigna a 'sinerrores' un TRUE, que "machaca" el posible resultado que pudiera haber en la variable 'sinerrores', luego evalúa si salir es FALSO, y en NINGÚN caso es posible que se detecte, ya que siempre evalúa una variable que está inicializada a TRUE. Solución: el programa nunca saldría. El compilador de C debe detectar los casos que no estén contemplados en la gramática del lenguaje. En este lenguaje está permitido esto y mucho más, como los famosos 'goto', que suelen ayudar a hacer ilegible el código. El compilador no detectaría error, y en las ejecuciones de prueba puede que tampoco detectáramos tal error, con lo cual ya tendríamos un "bug" vivito y coleando en nuestra "flamante" aplicación.

□ En Módulo-2: Para empezar, el Módulo, como el Oberon y otros lenguajes más formalizados, no te permite hacer una asignación dentro de una evaluación booleana, ni permite hacer 'goto', con lo cual no dejaría que nuestra aplicación se compilara sin generar un mensaje de error para que el programador se enterara de lo que ha sucedido.

Pero, como para todo en esta vida, en la programación no hay una piedra filosofal. Muchas de las rutinas gráficas de las magníficas 'demos' de los Amiga sólo pueden ser programadas en C o Ensamblador. Otras veces, no interesa tanto la fiabilidad como la velocidad de ejecución, como, por ejemplo, en un visualizador de animaciones. Otras veces es más necesario el ahorro de memoria que la fiabilidad, como, por ejemplo, en un juego para ordenadores con poca memoria. El código generado por lenguajes de alto nivel con una gran fiabilidad, como el Modula-2, suele ser más grande y lento que los generados en C, y, a veces, se necesita hacer "cosas raras" que el compilador no nos permite. Esta sección está orientada sobre todo a la programación fiable, con lo que será más fácil de implementar sobre Pascal o Módulo-2 que sobre C o Ensamblador.

La necesidad de la claridad en el código

La claridad en el código es, si cabe, imprescindible para el programador de hoy en día. Un listado bien anotado, con comentarios y con unas tabulaciones bien ajustadas que permitan una correcta lectura del código, es esencial. Un programador pasa más tiempo mejorando aplicaciones que programando nuevas, e, incluso en ese caso, se suele recurrir a "trozos" de anteriores programas o funciones que realizan acciones iguales o 'semejantes' a las que se necesitan. En un programa es tan importante su diseño como su mantenimiento. Un ejemplo claro de ello son los sistemas operativos actuales. Hay múltiples revisiones y mejoras del AmigaDOS, Finder, MSDOS, y lo mismo sucede con programas que tienen al menos un mínimo uso. Si la empresa en que se ejecuta un determinado software desea más prestaciones que las actuales, pero desea mantener el método de trabajo y las prestaciones existentes, se impone una revisión. Volvemos a lo de siempre: si el programa está bien diseñado, el nuevo programa será aún mejor y no tendrá problemas y si no, arrastrará problemas de una a otra versión hasta que no llegue a lo que se espera de él. Si el programador que lo codificó se encuentra dos años después con un listado críptico y laberíntico, con variables cuyo nombre no esté bien relacionado con su utilización, es más que probable que pierda gran cantidad de su tiempo en llegar a entender su propio listado para mejorarlo. Para ello es muy necesario que el texto esté bien estructurado y documentado, y que se guarden los algoritmos que se utilizaron en su diseño.

¿Qué es un algoritmo?

El algoritmo es, ni más ni menos, una expresión formalizada de lo que el programador tiene en mente. Es una representación de lo que se desea y lo que se obtiene, de sus entradas y salidas, de las

variables usadas y un estudio de los posibles estados en los que se puede encontrar un programa en su fase de ejecución. Normalmente, este algoritmo está escrito sobre papel, ya que es el modo más claro de entenderlo y consultarlo para el programador; es lo que diríamos un programa legible para una persona. El algoritmo no depende, normalmente, del lenguaje de programación utilizado. El algoritmo es el núcleo verdadero de un programa y es lo que el programador debe traducir a un lenguaje de programación para que la máquina pueda entenderlo. El algoritmo depende mucho de la persona que lo haya hecho, pero es muy recomendable utilizar algún sistema estándar para que más adelante podamos seguir entendiéndolo y que otras personas lo entiendan también.

Herramientas para la creación de programas correctos

La herramienta esencial para la creación de programas correctos es el dúo especificación-corrección. La idea en sí es bastante simple: un programador no sólo debe de crear un programa, sino que debe demostrar que éste funciona para todos los posibles casos. Se ha de programar de la misma manera en que un matemático formula sus métodos o resoluciones, mediante demostraciones.

Para reforzar esta afirmación diremos que, aproximadamente el programador es a las matemáticas lo que el arquitecto o ingeniero es a la física. Éstos construyen sus proyectos a partir de elementos físicos de la misma forma que un programador recoge los elementos matemáticos para construir sus programas. La informática es el medio por excelencia para representar y utilizar estas estructuras, ya que trabaja con ideas, con información y con objetos no físicos. Hasta la llegada de los autómatas programables sólo se podían representar a través de la música, geometría, etc., que eran medios que estaban muy limitados por elementos físicos como las dimensiones.

Así pues, la mayoría de los expertos en programación de hoy en día han llegado a la conclusión de que el programador debe demostrar sus programas. Para ello, se vale del método habitual: la lógica matemática. Los programas son ideas y ésta es la forma de evaluar las posibles interconexiones entre ellas.

La Lógica Matemática

Para quien no lo sepa, la lógica es la ciencia que nos ayuda a evaluar interconexiones entre unas ideas de las que partimos y que se suponen válidas. Esto suena complicado, pero la verdad es que la utilizamos inconscientemente:

"Juan me ha dicho que Pedro está con Luisa y Jesús me ha dicho que Luisa está en las Ramblas, luego Pedro debe estar en las Ramblas". En las frases anteriores tenemos dos premisas y una conclusión. Una premisa es un enunciado del que partimos y que se considera cierto:

"Pedro está con Luisa" y "Luisa está en las Ramblas" son las dos premisas de las que partimos y a partir de las cuales hemos de extraer una conclusión.

"Pedro está en las Ramblas" es una conclusión válida a la que llegamos tras operar inconscientemente las dos premisas anteriores con las leyes de la lógica.

La representación lógica de lo anterior sería ésta:

$$\begin{array}{l} P \div L \\ L \div R \\ P \div R \end{array}$$

La lógica intenta evaluar la validez de los razonamientos, no de los conocimientos que se aportan en sus premisas. Este razonamiento o interconexión de ideas seguiría siendo válido independientemente de que las premisas "Pedro está con Luisa" o "Luisa está en las Ramblas" sean o no ciertas. La lógica no está en absoluto ligada a las premisas que se formulan y de las que se parte. La siguiente frase sería evaluada como una interconexión válida entre ideas: "Sócrates era griego" ($S \div G$), los griegos eran verdes ($G \div V$), luego "Sócrates era verde" ($S \div V$). La forma de representar esto sería:

$$\begin{array}{l} S \div G \\ G \div V \\ S \div V \end{array}$$

Para aquellos que todavía encuentren extraña esta última afirmación, les diremos que también utilizamos este concepto inconscientemente. Si no, piensen que, seguramente, más de una vez han pronunciado frases del tipo. "Yo no te menté, a mí me dijeron que Pedro estaba con Luisa, y como Luisa estaba en las Ramblas, supuse que él estaría allí." Nuestro pensamiento en sí sería válido, aunque nuestro informador nos haya mentido, y como lo que la lógica evalúa es la validez de los razonamientos, éste sería dado por válido.

Las diferentes operaciones que se suelen encontrar en lógica son:

$\text{Op} \div \text{q}$ Condicional "Si p entonces q"

"Si es Sócrates, entonces es griego"

$\text{Op} \wedge \text{q}$ Conjunción "p y q"

"Sócrates es griego y Sócrates es verde"

$\text{Op} \vee \text{q}$ Disyunción "p o q"

"Sócrates es griego o Sócrates es verde"

$\neg \text{p}$ Negación "no p"

"Sócrates no es griego"

Hay otras operaciones lógicas como las variantes de éstas (disyunción negada, conjunción negada, etc...), o como los signos $\forall$ (para todo) y $\exists$ (existe).

Vamos a dar un último ejemplo de la potencia de la algorítmica metódica, aunque ésta no sea su sección. Hemos dicho que mediante ella podemos crear programas que podamos especificar mediante operaciones

lógicas. ¿Recuerdan las definiciones matemáticas del límite o las derivadas, todas ellas llenas de símbolos como el $\forall$ o el $\exists$? ¿Cómo lo implementarían? Estas operaciones lógicas son las mismas que se encuentran en los condicionales de cada uno de los lenguajes de programación que tenemos al alcance, por lo cual, aprenderlas no es ninguna carga extra para cualquier persona que vaya a programar. Más adelante daremos repases de lógica más avanzada. De momento, y para no tener que rebuscar en libros, las tablas de las operaciones booleanas elementales son éstas: (V =verdad y F =falso)

a	b	aAb	a v b	a -> b	$\neg$ a
F	F	F	F	?	V
F	V	F	V	?	V
V	F	F	V	F	F
V	V	V	V	?	F

Discusión sobre el condicional lógico

A estas alturas ya se habrá preguntado a qué se deben los interrogantes en la expresión del condicional. El condicional o "implica" sólo relaciona dos premisas con una conexión muy débil. 'a' "implica" 'b' ($a \rightarrow b$) nos dice que en caso de que 'a' sea verdadero, 'b' ha de serlo también. No nos dice nada de los casos en que 'a' sea falso (casos primero y segundo). En el caso en que 'a' sea verdadero, y 'b' falso, se observa que contradice la definición del "implica", es decir, 'a' es verdadero, pero no es falso. Éste es el único caso en el que se puede opinar sobre la veracidad o no del enunciado. En el caso que resta no podemos decir nada. Como pequeño ejercicio mental, determínelo usted mismo. Como ayuda, piense en lo siguiente: "Sócrates es europeo y es griego, pero eso no quiere decir que el hecho de que Sócrates sea europeo implique que sea griego"

Especificación

La especificación consiste en usar la lógica para expresar el estado lógico de salida en un programa y el estado lógico en el que debe finalizar el mismo, de forma que se nos garantice que el programa es correcto. En términos mundanos, lo que tenemos y lo que queremos, de lo que partimos y a lo que llegamos, en un algoritmo.

Nunca dos programas que realicen tareas diferentes pueden tener la misma especificación, y nunca una especificación puede dar lugar, salvo por error, a dos programas diferentes. Por ejemplo, un programa que calcule una división de dos números tendría una especificación determinada, y otro que intercambie dos variables, otra. Aquí presentamos un esquema un tanto informal de los pasos que sigue la mente de un programador antes de la realización de la especificación.

Ideas previas a la realización de un programa

Idea de la que partimos: Tenemos un numerador entero cualquiera 'x' y un denominador entero diferente de 0 llamado 'y'.

Idea a la que llegamos: Queremos un entero

'c' que sea el cociente de la división 'x DIV y' y otro 'r' que sea el resto.

Evolución de la idea: 'c' ha de ser tal que 'y*c+r=x' y a su vez, 'y*(c+1)' ha de ser mayor que 'x', y 'r' ha de ser menor que 'y' (definición matemática de la división entera).

Especificación:

Precondición: Expresión lógica de lo que tenemos al comenzar la ejecución del programa:

{ x e y son Enteros, y!=0 }

Postcondición: Expresión lógica de lo que tenemos al acabar el programa:

{ c y r son Enteros, y*c+r=x, y*(c+1)>x, r<y }

Ejemplo sencillísimo: Una función que intercambie el valor de x e y. El programa sería algo así:

```
Funcion cambio (entrada: x,y son enteros
salida: x,y son enteros)
{Precondición: x e y son Enteros, x = X,
y = Y}
VAR: comodin es entero;
comodin:=x;
x:=y;
y:=comodin;
{ Postcondición: x e y son Enteros, y = X,
x = Y}
retorna x,y;
finfuncion
```

El programa de la división de enteros quedaría así:

```
Función DIV (ent: x,y enteros sal:c,r enteros)
(* Función para calcular una división entera *)
{P: x e y son Enteros, y!=0}
c:=0;
r:=x;
mientras (r>=c) hacer
  c:=c+1; (* Vamos incrementando el
cociente hasta llegar al deseado *)
  r:=r-c; (* Decrementamos el resto
hasta que éste sea menor que el dividendo *)
finmientras
{ Postcondición: c y r son Enteros, y*c+r=x,
y*(c+1)>x, r<y}
retorna c,r;
finfuncion
```

Para que se vaya familiarizando con la especificación, he aquí la precondición y postcondición de otra función:

- Máximo de dos números:

{Pre: x e y son Enteros}

{Post: (x=>y)->(M=x), (y=>x)->(M=y)}

Pregunta: ¿Qué valor toma si x=y?

Respuesta aplastante: Da igual, ya que x=y.

La demostración

La demostración consiste en la verificación paso a paso de una función mediante el análisis de sus estados inicial, final e intermedios, y el estudio de cada una de las transformaciones de estos estados partiendo desde el inicial, pasando por los intermedios, y llegando al final.

En términos más comunes, consistiría en demostrar que el programa funciona perfectamente, verificando todos y cada uno de los estados intermedios. La Precondición y la Postcondición son los estados inicial y final. Saltando de uno a otro se puede demostrar que el algoritmo es correcto. A la precondición la llamaremos P, a la postcondición Q, y a los pasos intermedios P1, P2, P3, P4, etc. Para

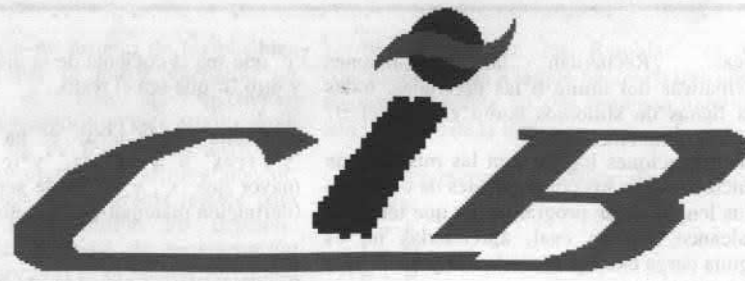
empezar, un ejemplo sencillísimo, el del cambio:

```
Función cambio (ent: x,y son enteros sal:x,y
son enteros)
{P: x e y son Enteros, x = X, y = Y}
VAR: comodin es entero;
comodin:=x;
{P1: comodin es Entero, comodin = x}
x:=y;
{P2: x e y son Enteros, x = Y, y = Y, comodin
es Entero, comodin = x}
y:=comodin;
{P3: x e y son Enteros, x = Y, y = X, comodin
es Entero, comodin = x}
{Q: x e y son Enteros, y = X, x = Y}
retorna x,y;
finfuncion
```

En el ejemplo se va viendo cómo se modifica la Precondición mediante las asignaciones hasta llegar a la Postcondición y queda demostrado realmente que el programa funciona.

OBSERVACIÓN FINAL: A simple vista, el desarrollo de funciones o programas verificados parece un poco tedioso, pero la verdad es que, con la práctica, es el mejor método de desarrollo de funciones. Si el lector observa con atención las Postcondiciones, se dará cuenta de que no son sino las definiciones o fórmulas matemáticas que nos encontramos en los libros de cálculo, física, química, etc. Al principio, no negamos que es mucho más rápido programar de la forma tradicional, pero esto tiene sus inconvenientes, como, por ejemplo, que está muy limitada por la "genialidad" del programador. Ha sido muy demostrado en los libros de programación que el programar "chupuza" vale para las cosas pequeñas y fáciles, pero se constata el hecho de que, ante una función complicada, el programador no tiene conocimientos ni sistemas fiables para "atacarla", sobre todo si es una función recursiva (que se llama a sí misma). En la programación metódica se dan herramientas para que, siguiendo paso a paso, mecánicamente, y partiendo de una fórmula matemática, se llegue siempre a un programa, más o menos efectivo y correcto que satisfaga las expectativas. Da, a su vez, mecanismos para transformar cualquier función recursiva (mucho más sencillas de lo que parecen), a la función iterativa (bucle) más optimizada que se pueda encontrar, y que sea de demostrada corrección, etc... Ya no se depende de que no se tenga un buen día ni de que no se sea un genio, la función sale sin problemas y, con la práctica, sin esfuerzo.

Además, para remarcar más los méritos de una programación correcta, existen métodos para "derivar" una función, es decir, para "coger" una especificación y de ella "sacar", mediante unos pasos siempre mecánicos e idénticos, una función que ya queda demostrada automáticamente. Pero esto se verá en próximos artículos. En el siguiente hablaremos de la notación que ya has estado usando en este capítulo, y mostraremos métodos de corregir los diferentes bucles, condiciones, etc... Incluiremos códigos fuente en distintos lenguajes, como Modula-2, Pascal y C, de los ejemplos que vayan introduciéndose en el curso. ●

**ACELERADORAS**

SUPRA TURBO 28 A2000	34.900 PTAS
SUPRA TURBO 28 A500	34.900 PTAS

DISQUETERAS

DISQUETERA INTERNA A500	12.900 PTAS
DISQUETERA EXTERNA TODOS AMIGA.	12.900 PTAS
DISQUETERA INTERNA HD A4000	24.900 PTAS

CHIPS AMIGA

AGNUS 1 MB 8372A	8.900 PTAS
CIA 8520	5.000 PTAS
ROM 2.05	8.900 PTAS
ROM 3.1 + WORKBENCH 2.1	24.900 PTAS
DENISE ECS	8900 PTAS

DISCOS DUROS

HD 250 MB 3.5" IDE PARA A1200	34.900 PTAS
HD 345 MB 3.5" IDE PARA A1200	37.900 PTAS
HD 420 MB 3.5" IDE PARA A1200	39.900 PTAS
HD 250 MB 3.5" SCSI PARA A2000	55.900 PTAS

AMIGA 1200
57.900 pts

GENLOCKS (ELECTRONIC DESIGN)
57.900 PTAS
69.900 PTAS
119.900 PTAS
129.900 PTAS
GENLOCK PAL
GENLOCK Y/C
GENLOCK SIRIUS
GENLOCK NEPTUNO

AMPLIACION DE MEMORIA

SIMM 4 MB 32 BIT A1200/A4000	29.900 PTAS
TARJETA A1200 4 MB RAM (ZYDEC)	44.900 PTAS

VARIOS

LAPIZ-RATÓN ÓPTICO	5.900 PTAS
ADAPTADOR 2 JOY/RATÓN	2.900 PTAS
BOOT SELECTOR ELECTRÓNICO	1.520 PTAS
MONITOR MULTYSINC "MICROVITEC"	89.900 PTAS
DUPLICADOR SALIDA VÍDEO	6.900 PTAS
FILTRO CRISTAL	1.900 PTAS
DISCOS 3.5" DD	50 PTAS
DISCOS 3.5" HD	100 PTAS
FUNDA SAFESKIN A2000/A4000	1.500 PTAS

CABLES

CABLE 44 PIN A1200	2.000 PTAS
CABLE ADAPT. HD 3.5" A1200	5.000 PTAS
CABLE EUROCONECTOR	5.000 PTAS

DIGITALIZADORES

FRAMEMACHINE & FM -PRISMA24 (Incluye Vídeo-Demo)	119.900 PTAS
VIDI 12 AGA	24.900 PTAS
VIDI 12 AGA RT	34.900 PTAS

CONTROLADORAS DISCO DURO
GOLEM SCSI A2000 9.900 PTAS
OCTAGON 2008 IDE A2000 19.900 PTAS
OCTAGON 2008 SCSI A2000 28.900 PTAS
FASTLANE Z3 A4000 59.900 PTAS

Otra forma de componer música con Amiga.



Santi Maspons nos inicia en el sistema más utilizado por los profesionales a la hora de crear música con el ordenador: EL INTERFAZ MIDI.

El atractivo mundo del MIDI relaciona campos tan sugerentes como la música y la informática. El primero, relacionado con las humanidades y las artes y el segundo, con la tecnología y el proceso de información. Ambos campos coinciden en que, al fin y al cabo, las notas musicales son información susceptible de procesar. Este vínculo aparece constantemente en el concepto multimedia, donde se aglutinan la información, la imagen y el sonido. Aunque para los usuarios de Amiga este concepto es algo natural y cotidiano, no ocurre lo mismo en otras plataformas.

MIDI es un interfaz digital de instrumentos musicales. La palabra en sí corresponde a las siglas en inglés de "Musical Instrument Digital Interface". Básicamente es una máquina de proceso digital que permite conectar varios instrumentos musicales digitales, ordenadores o secuenciadores de ritmos. Cada instrumento u ordenador dispone de circuitería para posibilitar la comunicación en el sistema. El Amiga dispone de una gran gama de interfaces MIDI para todos los presupuestos, desde menos de 5.000 pts. hasta más de 25.000 pts. El precio no determina la calidad del interfaz, sólo las prestaciones y comodidades que nos ofrece. El que usamos en nuestra configuración ronda las 9.000 pts (fig.1). Dispone de una entrada (IN) para un instrumento, salidas (OUT) para dos instrumentos y dos copias (THRU) más de la entrada. También nos ofrece un conector serie con la posibilidad de conmutarlo de MIDI a SERIE para no tener que desenchufarlo cada vez que usemos el puerto serie con un modem o impresora.



1. Muestra del interfaz midi externo, utilizado por Santi. Este modelo puede conectarse a cualquier Amiga.

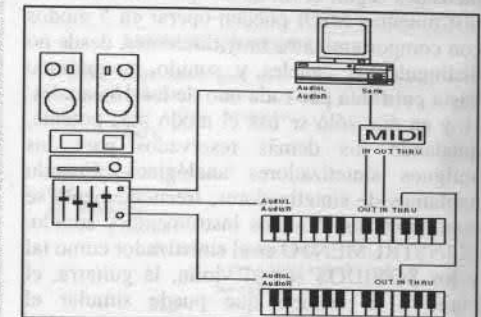
Los instrumentos musicales más frecuentemente usados en entorno MIDI, como los teclados musicales, sintetizadores de sonido o samplers llevan ya incorporado el interfaz. Con estos elementos, un programa de control MIDI si usamos ordenador y los cables necesarios, se pueden construir sistemas configurados para cada necesidad. Aunque en este artículo nos referiremos específicamente a la conexión entre el Amiga y dos teclados, caben todas las combinaciones. Se pueden conectar en cadena varios teclados de forma que un músico interprete una pieza musical en su teclado favorito y cada teclado reproduzca las notas que le sean indicadas. Se puede utilizar cualquier ordenador (con salida serie), pero como nos tiene acostumbrados, Amiga va a ofrecernos las ventajas de poder usar sus capacidades sonoras y sobre todo de trabajar en su entorno multiproceso.

CONEXIÓN DEL EQUIPO

La potencia del interfaz MIDI se basa en el envío de mensajes y NO de sonido, razón por la que en nuestro sistema habrá un cableado para señales MIDI y otro para el sonido. Por el sistema MIDI circula información de las teclas pulsadas, los sonidos que seleccionamos en el instrumento, el volumen, el tempo, otros comandos y señales de sincronización. No todos los instrumentos con interfaz MIDI reconocen el potente conjunto de mensajes; unos no aceptarán un cambio en el volumen u otros sólo dispondrán de un sonido y no tendrá sentido cambiarlo. El tipo de mensajes que reconoce el instrumento estará indicado en las especificaciones MIDI de su manual de usuario.

El interfaz MIDI transmite la información de forma secuencial bit a bit en octetos con un bit de inicio al principio y otro de parada al final para protocolo. Funciona a una velocidad de transmisión de 31250 bits por segundo (baudios). Suponiendo que la pulsación de cada nota se codifique en 30 bits (3 octetos más los bits de protocolo), podemos tocar más de 1000 notas por segundo, prestaciones más que suficientes para la mayoría de los músicos. Los cables por los que circulan los mensajes MIDI son unidireccionales, es decir, habrá una conexión de entrada (IN) y otra de salida (OUT). Opcionalmente, hay una tercera conexión llamada a través (THRU), que no es más que una copia de la entrada recibida al instrumento. Esta conexión extra nos permitirá copiar la señal que recibamos en un teclado hacia la entrada de otro y ésta a otro, etc... pudiendo así encadenar los instrumentos. La configuración que comentamos en este artículo corresponde a la figura 2 y consta de un Amiga

con un programa secuenciador de música que maneje MIDI, un interfaz MIDI, dos teclados y un equipo de amplificación.



2. Esquema de conexiones de la configuración utilizada. Se podrían añadir cajas de ritmos, módulos, etc ...

Veamos primero la conexión MIDI. Sin prestar mucha atención, podríamos conectar la salida del ordenador a la entrada del primer teclado y la salida de éste a la entrada del segundo. Sin embargo, esta configuración no permite controlar el segundo teclado desde el Amiga. Los mensajes del ordenador van al primer teclado, éste los interpreta y por su salida sólo envía las notas o controles que pulsamos. El segundo teclado sólo recibirá estas segundas señales y no las que enviábamos desde el programa secuenciador. La figura 2 muestra la conexión correcta en cadena. Si en lugar de conectar la salida del primer teclado, colocamos la conexión a través (THRU) a la entrada del segundo, los dos teclados recibirán los mensajes del ordenador. Las señales que reciba el primer teclado por la entrada son copiadas a la conexión a través y enviadas al segundo teclado.

Las conexiones de audio nos servirán para oír las composiciones a través de un equipo amplificado y para registrarlas en una cinta magnética. Conectaremos la salida de sonido del Amiga y de los teclados a una mesa de mezclas y ésta a un amplificador con altavoces y magnetoscopio. En esta configuración, el Amiga puede realizar las funciones de secuenciador de ritmos, de instrumento generador de sonidos y de editor de todos los parámetros de la composición musical.

Los dos teclados interpretarán las notas y mensajes que envíe el ordenador, pero sólo el primero nos permitirá enviar notas debido a que el interfaz MIDI sólo tiene una entrada. Para que los dos teclados pudieran enviar notas simultáneamente, necesitaríamos un fusionador MIDI. Conectaríamos las dos salidas MIDI de

los teclados a las entradas del fusionador y la salida de éste a la entrada del MIDI del ordenador. No obstante, con un pequeño truco podremos trabajar con los dos teclados sin el fusionador en nuestra configuración. Introduciremos las notas con el primer teclado y en el ordenador las modificaremos para que las interprete sólo el segundo. Aunque pueda parecer complicado, es una tarea muy simple gracias a la forma en que se codifican y envían los mensajes MIDI.

MODOS Y CANALES

La norma MIDI estándar considera 16 canales a través de los cuales se pueden enviar mensajes. Los canales se codifican y se implementan en un solo cable. No hace falta un cable para cada canal. Una vez instalado el sistema, no es necesario cambiar los cables. Cada instrumento puede interpretar los mensajes según el modo en que funcione. Los instrumentos MIDI pueden operar en 5 modos con comportamientos muy diferentes, desde no distinguir los canales y sonido monofónico hasta polifonía por cada uno de los 16 canales. Hoy en día, sólo se usa el modo más potente, quedando los demás reservados para los antiguos sintetizadores analógicos. Cuando hablamos de sintetizadores, frecuentemente se confunden los términos instrumento y sonido. El INSTRUMENTO es el sintetizador como tal y los SONIDOS son el violín, la guitarra, el piano o el bombo que puede simular el instrumento sintetizador. Muchas veces llamamos teclado al instrumento y instrumentos a los sonidos que éste puede sintetizar, se entenderá según el contexto. Se podría considerar a los sonidos del instrumento como instrumentos virtuales. Hablaremos de POLIFONIA cuando varias notas puedan activarse simultáneamente y evidentemente suenen todas. El número de notas que pueden sonar a la vez se llama también número de VOCES. Tendremos MONOFONIA cuando al activar una nota se desactive automáticamente la anterior, sonando siempre una sola nota como máximo. Otro concepto que aparece en el entorno MIDI es OMNI. Omni desconectado (omni off) indicará que se consideran los canales MIDI y se reciben o envían mensajes por diferentes canales. En cambio, omni conectado (omni on) ignora los canales, considerándolos como un único canal. Vistos estos conceptos, veamos las posibilidades que ofrece cada modo de funcionamiento de los instrumentos que incorporan MIDI:

MODO 1 (omni on / poly): Considera todos los canales como uno. Podemos enviar mensajes al instrumento por cualquier canal y los aceptará. Este modo no permite diversos instrumentos simultáneos pero nos permite polifonía. Pueden sonar varias notas a la vez pero de un único sonido. Evidentemente, si el instrumento (teclado) tiene diversos sonidos, podemos cambiar el sonido para cada fragmento de la composición.

MODO 2 (omni on / mono): Tampoco distingue los canales pero además es monofónico. La activación de una tecla provoca la desactivación de la anterior. Es el modo más simple, sólo suena una nota a la vez pero también podemos cambiar de sonido si el instrumento lo permite.

MODO 3 (omni off / poly): Este modo distingue los canales. Podemos poner un sonido diferente en cada canal y además en polifonía. Es el modo más utilizado actualmente por los profesionales. Por ejemplo, podemos tocar un violín polifónico por el canal 2, una guitarra polifónica por el 3, un piano a varias voces por el 4 y un ritmo de batería por el 10, todos ellos simultáneamente y sincronizados.

0xxxxxxx	(Dato)
1000cccc	0nnnnnnnn 0vvvvvvv (Desactivación nota) (nota) (velocidad)
1001cccc	0nnnnnnnn 0vvvvvvv (Activación nota) (nota) (velocidad)
1010cccc	0nnnnnnnn 0vvvvvvv (Activación nota) (nota) (presión)
1011cccc	0ccccccc 0vvvvvvv (Cambio de Control) (control) (valor)
1011cccc	00000001 0vvvvvvv (Control) (Rueda de modulación) (valor)
1011cccc	00000010 0vvvvvvv (Control) (Controlador de soplo) (valor)
1011cccc	00000100 0vvvvvvv (Control) (Pedal) (valor)
1011cccc	00000101 0vvvvvvv (Control) (Tiempo de portamento)
1011cccc	00000110 0vvvvvvv (Control) (Botón de entrada de datos) (valor)
1011cccc	00000111 0vvvvvvv (Control) (Control de volumen principal) (valor)
1011cccc	00001000 0vvvvvvv (Control) (Balance) (valor)
1011cccc	00001010 0vvvvvvv (Control) (Pan) (valor)
1011cccc	01000000 0vvvvvvv (Control) (Pedal de sostenido) (valor)
1011cccc	01000001 0vvvvvvv (Control) (Portamento) (valor)
1011cccc	01000010 0vvvvvvv (Control) (Sostenido) (valor)
1011cccc	01000011 0vvvvvvv (Control) (Pedal soft) (valor)
1011cccc	01011100 0vvvvvvv (Control) (Profundidad de trémolo) (valor)
1011cccc	01011101 0vvvvvvv (Control) (Profundidad de coro) (valor)
1011cccc	01011110 0vvvvvvv (Control) (Profundidad de celeste) (valor)
1011cccc	01011111 0vvvvvvv (Control) (Profundidad de fase) (valor)
1011cccc	01100000 0vvvvvvv (Control) (Incremento entrada datos) (valor)
1011cccc	01100001 0vvvvvvv (Control) (Decremento entrada datos) (valor)
1011cccc	01100010 0aaaaaaa (Control) (Parámetro no registrado) (valor)
1011cccc	01100011 0bbbbbbb (Control) (Parámetro no registrado) (valor)
1011cccc	01100100 0aaaaaaa (Control) (Parámetro registrado) (valor)
1011cccc	01100101 0vvvvvvv (Control) (Parámetro registrado) (valor)
1011cccc	01110100 00000000 (Control) (Control local) (desactivado)
1011cccc	01110101 01111111 (Control) (Control local) (activado)
1011cccc	01110110 00000000 (Control) (Desactivación todas notas)
1011cccc	01110111 00000000 (Control) (Desactivación modo omni)
1011cccc	01111100 00000000 (Control) (Modo omni)
1011cccc	01111101 00000000 (Control) (Modo monofónico) (número de canales)
1011cccc	01111110 00000000 (Control) (Modo polifónico)
1100cccc	0ppppppp (Cambio de programa) (programa)
1101cccc	0ppppppp (Pulsación global de canal) (presión)
1110cccc	0aaaaaaa 0bbbbbbb (Modificación pitch) (bajo) (alto)
11110000	(Inicio sistema exclusivo)
11110001	(Indefinido)
11110010	0aaaaaaa 0bbbbbbb (Posición canción) (bajo) (alto)
11110011	0ccccccc (Selección de canción) (número de canción)
11110100	(Indefinido)
11110101	(Indefinido)
11110110	(Requerimiento de entonación)
11110111	(Fin sistema exclusivo)
11111000	(Reloj de Temporización)
11111001	(Indefinido)
11111010	(Inicio)
11111011	(Continuación)
11111100	(Parada)
11111101	(Indefinido)
11111110	(Espera activa)
11111111	(Inicialización del sistema)

Tabla 1. Resumen de todos los mensajes que implementa MIDI. Aparece la codificación en binario de cada comando y el número de datos que precisa.

MODO 4 (omni off / mono): Este modo permite el uso de canales diferentes, pero por cada canal sólo sonará una nota a la vez. Es muy útil en tareas de secuenciación de ritmos. Se pueden enviar las notas de cada elemento de una batería por un canal diferente. Esta forma de trabajar ocupa muchos canales y puede utilizarse en una banda donde sólo se emplee el MIDI para una base rítmica y haya otros instrumentos no controlados vía MIDI como una guitarra clásica, una flauta o un coro.

MODO 5 (multi): Modo no implementado en todos los instrumentos. Permite la asignación dinámica de la polifonía

de cada canal. Se puede programar el número de voces que pueden sonar por cada canal. Hay que tener en cuenta el grado de polifonía (número de voces) que permite el instrumento.

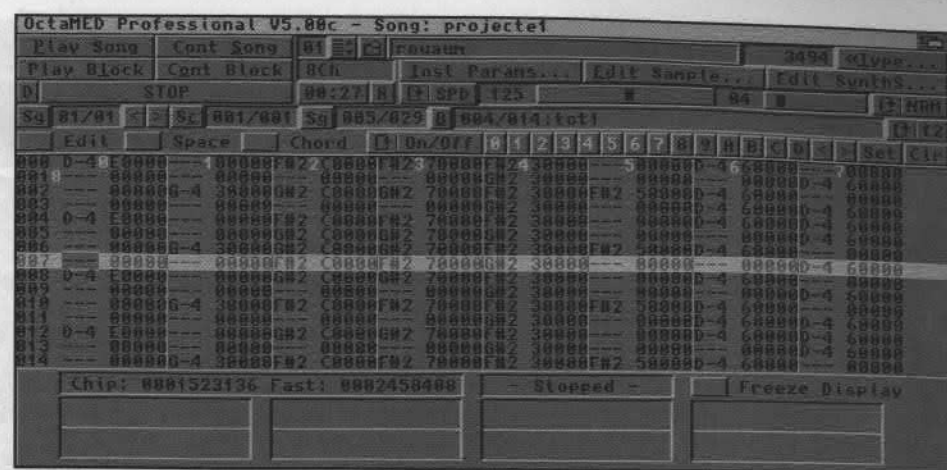
LOS MENSAJES MIDI

Existen cuatro tipos de mensajes MIDI: los mensajes a través de canal, los comunes a todo el sistema, los exclusivos de

canal es un octeto con el bit más significativo igual a 1, seguido por tres bits mmm que codifican el comando y por cuatro bits cccc que codifican el canal por el que se envía el mensaje. La codificación del canal va desde 0000 (0) hasta 1111 (15), aunque se habla de los canales 1 al 16. El código 0000 indicará el canal 1, el 0001 el canal 2 y así sucesivamente hasta el 1111 que indicará el 16. Los octetos de datos SIEMPRE comienzan con un 0. Los 7 bits restantes codifican el dato con unos valores posibles entre 0 y 127. La tabla 1 resume todos los mensajes que implementa MIDI. En ella aparece la codificación en binario de cada comando y el número de datos que precisa.

Comentaremos algunos de los mensajes que aparecen en el uso de programas de control MIDI. El programador de aplicaciones precisará conocer el funcionamiento detallado de todos ellos, pero con un programa adecuado como MUSIC-X, BARS & PIPES o incluso OCTAMED, un músico podrá trabajar en el entorno MIDI sin conocer la existencia de las señales.

Los comandos de activación y desactivación de notas tienen como operandos la nota codificada en el primer octeto de datos y la velocidad de pulsación o volumen de la misma en el segundo. La interpretación de este comando puede variar de un teclado a otro. Se puede interpretar la velocidad como la velocidad de ataque de la nota o como el volumen con que debe sonar. Los mensajes de cambio de control sirven para cambiar parámetros diversos del instrumento como el volumen general, el sostenido de las notas o el modo en que opera el instrumento. Cada fabricante indica, en su manual de usuario, los controles que soportan MIDI. El cambio de programa se interpreta usualmente como un cambio de sonido del teclado, por ejemplo para escoger el violín o el piano. El simple hecho de presionar una tecla y soltarla se traduce en dos mensajes de 3 octetos cada uno. Al pulsar la tecla se genera un primer mensaje que indica la activación de la nota con una determinada velocidad. Cuando soltamos la tecla provocamos otro mensaje que indica la desactivación de esa nota a una determinada velocidad. Las velocidades de activación y desactivación son independientes. Una velocidad de activación nula provoca el mismo efecto que desactivar una nota, aunque los



3. OctaMED Professional V5.00 es uno de los programas que sirven para realizar composiciones MIDI. Es apropiado tanto para las composiciones MIDI como para realizar módulos clásicos de Amiga, compatibles con Protracker, Soundtracker, etc...

circuitos del instrumento probablemente estén trabajando. Otra forma de desactivar la nota es a través del comando desactivar todas las notas.

Todos estos mensajes circulan por el sistema a tiempo real, es decir, cuando se producen, se interpretan. Si tenemos un grupo de mensajes preparados en el ordenador y los mandamos vía MIDI (puerto serie) a un instrumento, es probable que en el instrumento suene algo parecido a un chillido. Todas las notas han sonado a la vez. La razón es que no hemos secuenciado correctamente los mensajes. Cada nota tiene que sonar en un instante concreto y durante un determinado tiempo. Cada mensaje debe ser enviado en su preciso momento para que aparezca eso que llamamos música. Las funciones de sincronización las realizará el programa secuenciador que usemos en el ordenador. Los programas que utilizaremos normalmente nos mostrarán los datos de los mensajes en el sistema decimal o hexadecimal, por lo que habrá que hacer las conversiones convenientes si usamos la TABLA 1. La información de activación y desactivación de notas y otros mensajes, suelen estar representados en un pentagrama o un símil que incluye una trama de tiempo y muestra el momento en que se produce cada mensaje.

MUSIC-X Perf: Extermini1.total

BEGIN

END

CUE1

CUE2

CUE3

CUE4

SET

II PAUSE

PLAY

RECORD

STOP

Clock: 0036.01.023

Time: 00:00:57.27

Memory: 316823 Events

STORE

EDIT

DELETE

PREVIEW

SEQUENCER

Bar Seq

Bar Seq

36 Control

2 baix1

14 Yamaha1

6 cors 1

Seq	Mem	Bars	Channels	CST	Time	Sequence Name	Out	<<	>>
001	1292	181	11	-S-	Rel	Control	Ex1	00000	
002	412	4	2	---	Rel	baix1	Ex1	Off	
003	892	4	10	---	Rel	bats1	Ex1	Off	
004	842	4	10	---	Rel	bats2	Ex1	Off	
005	902	5	10	C--	Rel	bats1 desfase	Ex1	Off	
006	342	36	4	---	Rel	cors 1	Ex1	Off	
007	292	1	10	---	Rel	bats1 to variation	Ex1	Off	
008	132	16	1	---	Rel	Yamaha1	Ex1	Off	
009	652	16	3	---	Rel	digi breath 1	Ex1	Off	

4. MUSIC-X es la opción más barata y además es más que adecuado para el principiante. Permite un control completo del entorno MIDI así como de los sonidos muestreados (samples) del Amiga.

LISTADO NÚMERO 1:

```
/*LEER MIDI.REXX por Santi Maspons 1994 */
if -open('midi','ser:','Read') then do
  say 'Ser: ocupado...'
  exit
end
say 'Puerto serie:'
c=' '
do forever
  x=254
  do while (x=254)
    c=readch('midi',1)
    x=c2d(c)
  end
  say x
end
```

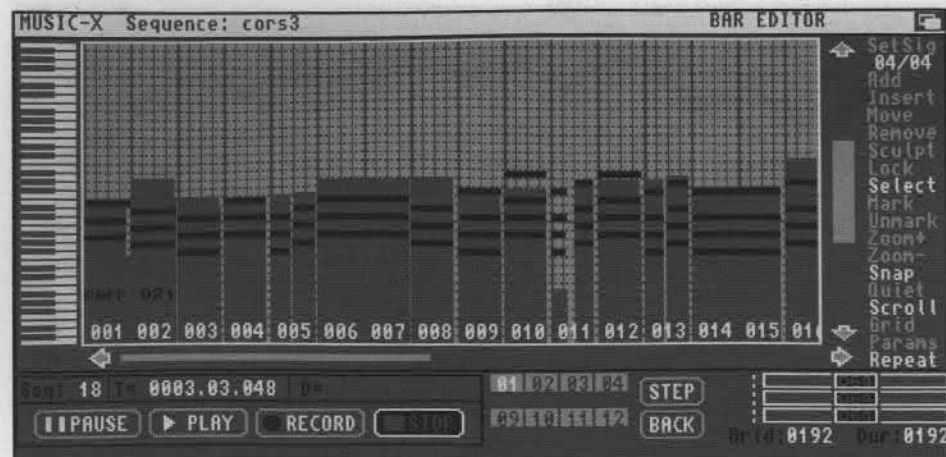
PROGRAMACION MIDI

Para comprobar que todo funciona, podemos ejecutar unas pequeñas utilidades en AREXX para consultar y enviar los mensajes MIDI. Para ejecutarlos sólo hay que editar un fichero ASCII (script) con el listado y ejecutar RX <nombre del fichero> desde un Shell. El primer programa (LISTADO 1) reserva el puerto serie y va mostrando en base diez los datos que circulan por el sistema. Para probarlo basta con ir tocando notas en el teclado o cambiar los sonidos y controles. Si por ejemplo, al variar un control en el teclado, el programa no muestra ningún valor, podemos suponer que la implementación MIDI de nuestro teclado no incluye el cambio de ese control. Podemos confirmarlo en el manual del instrumento. El programa espera hasta que reciba algún mensaje diferente de espera activa (254 en decimal o 11111110 en binario) y lo imprime en la pantalla. Para abortarlo basta con pulsar <CONTROL>+C en el Shell.

LISTADO NÚMERO 2:

```
/*ESCRIBIR MIDI.REXX por Santi Maspons 1994 */
if -open('midi','ser:','Write') then do
  say 'Ser: ocupado...'
  exit
end
say 'Usando SER: ...'
do forever
  x=readln(stdin)
  if x='' then break
  c=d2c(x)
  writech('midi',c)
end
```

El segundo programa nos permite enviar mensajes al teclado. Podemos probar con enviar 164, 60 y 64 para activar una nota y 132, 60 y 64 para desactivarla.

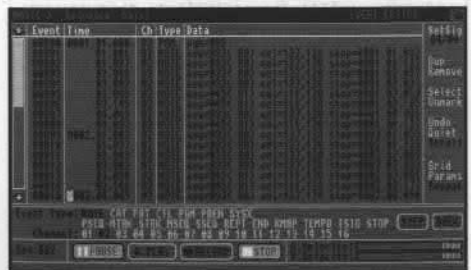


5. Todos los secuenciadores tienen la facilidad de modificar los datos una vez introducidos. En cada pista pondremos la notas compuestas de cada sonido.

TRABAJANDO CON MIDI

Normalmente trabajaremos con algún programa que nos secuencie los mensajes MIDI. El Amiga dispone de programas de música con características propias. BARS & PIPES puede incluso sincronizar animaciones de REAL 3D o presentaciones de gráficos con la composición musical. Los programas del tipo OCTAMED, PROTRACKER y OCTALYZER están especializados en los sonidos del Amiga. Para compatibilizarse con los canales MIDI, implementan 16 voces. Aunque el Amiga dispone de 4 voces, estos programas son capaces de hacer sonar 8 voces simultáneas en el ordenador y 8 voces más vía MIDI. OCTALYZER consigue 8 voces en el Amiga de forma ejemplar. Este último grupo de programas tiene unas cualidades perfectas si queremos trabajar con sonidos muestreados (samples) del Amiga pero no es muy fiel a la norma MIDI y no será la mejor opción para trabajar profesionalmente con sintetizadores u otros instrumentos MIDI. Nosotros hemos escogido MUSIC-X, un veterano programa para Amiga que permite un control completo del entorno MIDI así como el manejo de los sonidos del Amiga.

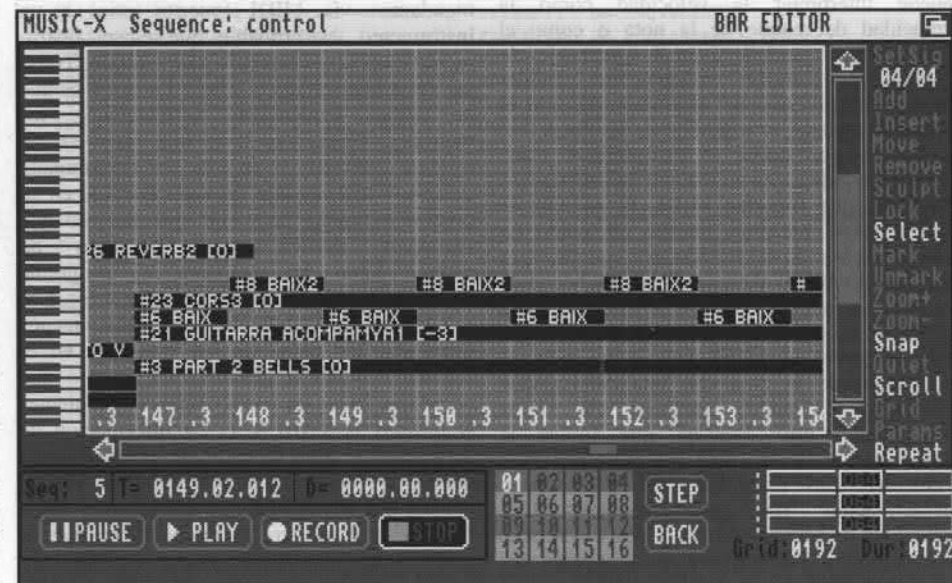
Music-X se presenta como un secuenciador de 250 pistas o secuencias (Figura 4). Se pueden componer las notas de cada sonido en una pista diferente (Figura 5). Cada secuencia puede tener las notas que queramos, incluyendo polifonía. Las notas se representan en MUSIC-X como barras de colores. Asignando un color a cada canal. Esta iconografía es muy intuitiva y flexible en su uso. Las notas que deben sonar juntas aparecen en columna. La longitud de la barra representa la duración de la nota. Las notas pueden activarse y desactivarse en cualquier momento.



6. MUSIC-X es ya un programa veterano, pero reúne cualidades que lo hacen perfecto para el aprendizaje del MIDI.

La resolución temporal de MIDI es muy alta y no se percibe. Este hecho marca una gran diferencia con los programas tipo OCTAMED, en los que sólo se puede activar una nota cada cierto tiempo.

Un método de trabajo para componer una pieza es comenzar por programar una secuencia de metrónomo en el ordenador y hacerla sonar con un sample tipo tic.



7. Una vez introducidas las notas, podremos modificar todos los parámetros para conseguir que nuestro Amiga simule toda una orquesta al completo.

Pondremos el programa de modo que pueda registrar los mensajes MIDI con algún dispositivo que indique RECORD o algo similar. El programa debería permitir grabar todos los canales MIDI por los que se le envíen mensajes. Algunos programas pintan las notas que se reciben por cada canal de colores diferentes. En el teclado podremos ir probando la melodía que nos rondaba por la cabeza. Si todo va bien, el programa va indicando de alguna forma que recibe los mensajes y los va guardando. Una vez finalizada la interpretación, el programa debería dejarnos guardar la secuencia tocada, ponerle un nombre, corregir, eliminar o añadir notas y insertar mensajes como cambio de sonido, o volumen. Se pueden ir grabando secuencias mientras escuchamos las grabadas

anteriormente de forma parecida a un grabador multipista, y así ir añadiendo efectos, coros y otras voces. Una vez tenemos todas las secuencias grabadas, podremos ponerlas en orden, cambiar el canal por el que se enviarán, cambiarlas de octava o incluso hacer que las interprete el Amiga con nuestros samples o sonidos preferidos. De esta forma, un solo músico puede componer y interpretar la totalidad de la pieza musical y controlar todos los instrumentos conectados. Si configuramos cada teclado para que reconozca unos canales MIDI e ignore el resto y cambiamos los canales por los que se envían las secuencias en el programa, podemos conseguir que las notas que habíamos introducido desde un teclado suenen en el otro. De esta forma solucionamos el problema de tener una única entrada en el MIDI. Mientras el Amiga y los dos teclados estén interpretando nuestra composición, podemos también tocar las notas que nos permita la polifonía del instrumento. De hecho ésta es la manera de poder hacer un concierto en vivo sin una banda, el músico toca las principales melodías, y el programa secuenciador los ritmos y acompañamientos. Este método de trabajo con MIDI permite la sincronización de la música con eventos exteriores para crear un gran espectáculo multimedia como logra, por ejemplo, Jean Michel Jarre en todos sus conciertos. Desde este artículo invito a todos los músicos

(profesionales o amateurs) que usen instrumentos susceptibles de usar MIDI a conectarlo a un Amiga. ●



8. Utilizando MUSIC-X podrás usar tus samples favoritos en lugar de los sonidos del instrumento.

AMIFORO

luz verde para tu Amiga...

Aunque algunos se lo salten a la torera, todos sabemos para qué sirve un semáforo... pero seguro que este no se lo saltará nadie, ya que Xim Montero nos fabricará un "Amiforo" que nos ayudará a comprender el control de montajes conectados al puerto paralelo.

En el Amiga, como en todos los ordenadores, existen puertos de expansión que permiten a nuestro equipo poderse ampliar y conectarse a periféricos externos como impresoras, modemis, otros ordenadores, etc...

Los puertos más comunes en todos los ordenadores son el paralelo, el serie, el bus de expansión y el de Joystick/Ratón.

El puerto paralelo, además de otros controles, lleva 8 líneas (cables) para comunicación tanto de entrada al ordenador como de salida (del ordenador al exterior). Las 8 líneas se asignan a un byte (8 bits) y cada línea puede tener dos estados eléctricos que se corresponden con los dos estados de los bits (1's y 0's).

El puerto serie, permite la comunicación a través de un solo cable, con lo cual los bits van uno detrás de otro (a diferencia del puerto paralelo, en el cual los datos viajan simultáneamente por los 8 cables). Los buses de expansión son conectores que están unidos directamente a las memorias y al procesador (CPU) y así podemos controlar directamente todo el ordenador desde el exterior (es el caso del Action Replay, o los típicos cartuchos de ordenadores C64, Spectrum, Consolas de videojuegos, etc).

El puerto de juegos permite la conexión del ratón, paddles, joystick, lápiz óptico, y, en general, cualquier dispositivo que se nos ocurra.

Para el artículo presente utilizaremos el puerto paralelo. La idea es montar un semáforo. Como tenemos que controlar varias luces simultáneamente debemos tener varias líneas de control.

En el puerto paralelo hay 13 líneas de comunicación de las cuales 8 son propiamente las de datos, 3 son líneas auxiliares también llamadas líneas genéricas de control "all purpose in/out control lines" (por ejemplo para que una impresora informe de que no tiene papel, etc.) y 2 son de control para asegurar que los datos han llegado a su destino. Las dos líneas de control últimas funcionan automáticamente, sin embargo en este montaje no las vamos a tener en cuenta. Las otras líneas

de control (las 3 líneas genéricas) no las vamos a usar tampoco. Usaremos únicamente las 8 líneas de datos.

Estas líneas pueden ser de entrada o de salida, y programables cada una de ellas independientemente de las demás. Cada aplicación concreta las configura a medida. No tienen por qué estar todas a IN (entrada) o todas a OUT (salida).

Una línea está a IN si permite que el ordenador detecte datos del exterior. Una línea está a OUT si permite al ordenador actuar sobre un dispositivo exterior.

Las líneas del ordenador son de comunicaciones. Esto significa que pueden soportar poca potencia (ni se os ocurra poner un motor directamente al puerto paralelo). A pesar de eso pueden llegar a iluminar unos pocos LED's (Light Emitting Diode). Un LED es un dispositivo que se ilumina con unos 10 miliamperios (Ver figura 7). Si ponemos 6 Leds (dos rojos, dos amarillos y dos verdes) tenemos un total de 6x10=60 miliamperios. El puerto paralelo puede soportar hasta 100 miliamperios (mA) (dato que se puede sacar de los libros que te dan cuando compras el Amiga, sección de conectores) así que todavía podemos encender los 6 a la vez si interesa. (Además, de hecho, no estarán los 6 a la vez en un funcionamiento normal, con lo cual consumiremos menos corriente).

¿COMO ENCENDER UN LED?

Para encender un LED el montaje típico es el de la figura 1, donde los 5 voltios son un valor que no tiene por qué ser fijo (una pila de 4.5 voltios sirve igual, una fuente de 15v

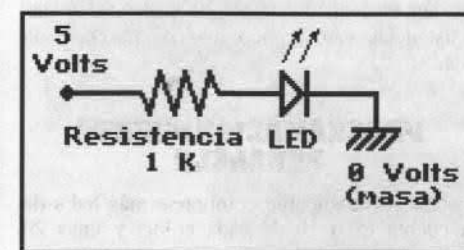


FIGURA 1. Circuito típico usado para encender el led. Se requiere un generador de 5 Voltios y, una resistencia.

también, una batería de coche (13.8v)...).

Pongo 5 voltios porque el Amiga entrega 5v en un pin (el 14) del puerto paralelo.

¡¡¡Atención usuarios AMIGA 1000!!! El puerto paralelo tiene otra configuración. Consultar esquema del manual para ver equivalencias. Estos voltios los vamos a aprovechar. La resistencia de 1K (o sea 1000 ohms, ver figura 8 para identificar los colores) limita la corriente. La ley de Ohm dice lo siguiente:

$$I \times R = V$$

donde 'V' es el voltaje en bornes de la resistencia y 'I' la intensidad de corriente (mide la cantidad de electrones) que circulan a través de dicha resistencia.

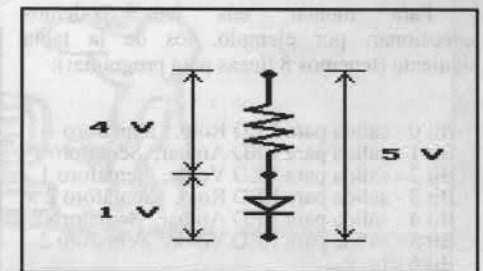


FIGURA 2.

Cuando el LED se enciende "cae" 1 Voltio en sus bornes o terminales, esto significa que sólo 4 van a caer en bornes de la resistencia (ver fig.2). Si ponemos una R de 1K (una resistencia de 1000 ohms) podemos saber la cantidad de corriente que circula así $I \times R = V \rightarrow I = V / R \rightarrow I = 4V / 1000 = 4mA$ (4 miliamperios). Estos amperios indican (o están relacionados) con el número de electrones que atraviesan la resistencia y que que obligatoriamente deben atravesar el LED para cerrar el circuito. Al atravesar el LED lo encienden.

Habíamos dicho que un led se enciende con 10 mA, sin embargo si sólo pasan 4mA como hemos calculado, se iluminará con menos intensidad pero suficiente para poder ser visible. En otros artículos veremos cómo dar potencia a las salidas del ordenador.

Si quisiéramos encender un poco más los LED's podríamos poner una resistencia más

pequeña para que la intensidad subiera. Por ejemplo si busco una intensidad $I=12\text{mA}$ tendría

$$I \times R = V \rightarrow R = V / I \rightarrow R = 4 \text{ V} / 12 \text{ mA} = 4 / 0.012 = 333 \text{ Ohms}$$

Con 12 mA se ilumina ligeramente más que con 10 mA. Un LED aguanta bastante (puede aguantar hasta unos 100mA) pero no tiene sentido forzarlo mucho porque se nos acabaría quemando. Un LED en condiciones normales (10 - 15 mA) tiene una vida aproximada de unos 50 años.

Los pines del puerto actúan como interruptores, de manera que, cuando hay un 0 lógico programado, el interruptor cierra el pin del conector a "masa" o sea cero voltios. Si hay un 1 lógico programado, el interruptor se abre y no pasa corriente. Ver figura 3.

Así que, podemos coger un pin que se corresponda con un bit de datos y conectarlo como en la figura 4.

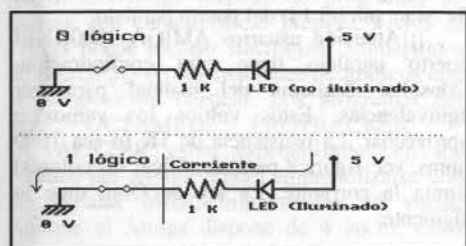


FIGURA 4.

Los pines de control que no usamos (general purpose lines) también actúan así.

Para montar seis leds podemos seleccionar, por ejemplo, los de la tabla siguiente (tenemos 8 líneas para programar):

- Bit 0 - salida para LED Rojo, Semáforo 1
- Bit 1 - salida para LED Ambar, Semáforo 1
- Bit 2 - salida para LED Verde, Semáforo 1
- Bit 3 - salida para LED Rojo, Semáforo 2
- Bit 4 - salida para LED Ambar, Semáforo 2
- Bit 5 - salida para LED Verde, Semáforo 2
- Bit 6 - libre
- Bit 7 - libre

Observar que los bits 6 y 7 de datos (de líneas de datos) quedan libres y por tanto podemos utilizarlos, por ejemplo, para poner dos interruptores como entrada.

Un bit de datos de entrada, a efectos electrónicos es como si no estuviera conectado y hubiera un "bloque espía" que observara si hay o no tensión (voltaje) en ese pin o pata del conector (ver figura 5)

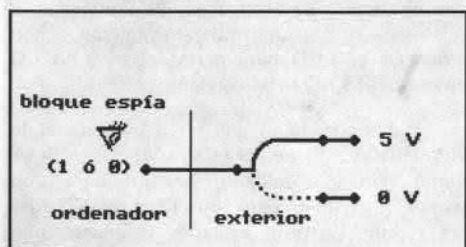


FIGURA 5.

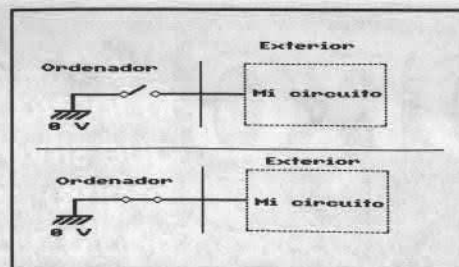


FIGURA 3.

Un montaje típico es el de la figura 6.

Si el interruptor está cerrado, la conexión es directa a masa, a cero voltios, y en el ordenador se detecta un 0 lógico (los 5 voltios quedan como "separados" por una resistencia que no influye).

Ahora bien, si el interruptor lo abrimos, el pin del ordenador no tiene NINGÚN contacto con masa, con los 0 voltios y sí en cambio

tiene un camino a 5 voltios a través de la resistencia. Como no hay ningún circuito cerrado (interruptor abierto, y el bloque espía es como si no tocara el pin del puerto) la corriente es cero porque los electrones no pueden circular por ningún camino, y por tanto, según la ley de Ohm, $IR=V$, si $I=0$ entonces $V=0$ en bornes de la resistencia. En la resistencia "caen" cero voltios, si por un lado la conectamos a 5 voltios, al otro lado tenemos también 5 voltios. Los 5 voltios llegan al pin con lo cual detectamos un 1 lógico. Para cada uno de los pines de salida del puerto, hay que replicar el circuito salida-resistencia-led, y obtendremos el esquema del montaje completo como el que está en la figura 9.

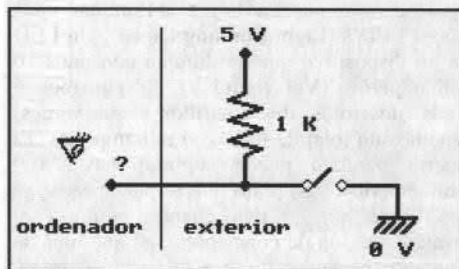


FIGURA 6.

Ya lo tenemos todo. Sólo falta comprar un conector D-25 para el puerto, como el de la figura 12, comprar los LED's, las resistencias y los interruptores, comprar una plaquita de circuito impreso y ponerse a soldar. El conector puede valer unas 100 pesetas, un led 10 pesetas (y si los pedimos de alta eficiencia o alta luminiscencia, brillan más con la misma corriente, costarán unas 30 pesetas), y las resistencias van a poco más de peseta cada una.

PROGRAMACION PUERTO PARALELO

Es recomendable comprarse más led's de la cuenta (5 ó 10 de cada color) y unas 20 resistencias de golpe (de 1K, pero si queréis, comprad ya de otros valores para experimentar por vosotros mismos).

Aun falta comentar cómo se programan los 1's ó 0's en los pins de salida (abrir o cerrar el interruptor a masa) y cómo se leen los 1's y los 0's de entrada (ver si hay o no tensión en un pin de entrada).

El puerto lo tenemos conectado internamente a los llamados CIA del ordenador. Los CIA son unos circuitos integrados (Chips) que el Amiga usa para controlar dispositivos y comunicarlos con el procesador. El Amiga lleva dos CIA (Complex Interface Adapter). Cada uno lleva dos puertos paralelos completos, de los cuales tres son para conexiones internas (con el Led del Power, con el botón del ratón, del Joystick, control de las líneas auxiliares del puerto serie, control unidad de disco, etc.). El puerto que queda libre, Commodore lo ha conectado por completo al puerto paralelo accesible en las conexiones externas.

Para programar el puerto paralelo, necesitamos programar el "puerto B" del llamado CIA Impar (existe el CIA Par y ahora no nos interesa). El "puerto A" controla el botón izquierdo del ratón y el botón del Joystick.

Para programar cada uno de los puertos de los CIAS debemos acceder a dos posiciones de memoria donde está mapeado el chip.

Para el CIA impar las cuatro direcciones de interés son:

- \$bfe001 - DRA
- \$bfe101 - DRB
- \$bfe201 - DDRA
- \$bfe301 - DDRB

Donde A y B son los nombres de los puertos (el paralelo es el B)

DR significa Data Register (Registro de datos) y contiene los datos que debemos leer o escribir.

DDR significa Data Direction Register (Registro de las direcciones en que van los datos). Un uno significa, en todos los ordenadores, una salida, y un cero es una entrada. Esto se hace así para que cuando encendamos el ordenador (Absolutamente todos los registros del ordenador a cero) las líneas sean de entrada y por tanto no puedan salir datos al exterior y provocar algún funcionamiento indeseado en algún circuito.

Así, si conectamos los bits 0 a 5 del puerto paralelo a los leds y el 6 y el 7 a los interruptores, necesitamos que de 0 a 6 sean salidas (1 en el DDR) y los bits 6 y 7 sean entradas (y por tanto 0 en el DDR).

Haremos, desde código máquina:

MOVE.b #00111111, \$bfe301 ;DDRb, observad que es .b (byte)

Ahora falta programar los datos. Encender el Led es cerrar el circuito, o sea, escribir un cero. Si quiero encender el rojo del semáforo 1 y el verde del semáforo 2, hay que encender los leds de los bits 0 y 5 y el resto apagados, es decir, poner a cero los bits 0 y 5 y poner a uno el resto. El valor que escribamos en los bits 6 y 7 no significa nada puesto que las líneas son de entrada y el bit que nosotros movemos como

MATERIAL UTILIZADO

Un conector D-25 MACHO (para soldar en placa), como el de la figura 11, 6 LED's (2 rojos, 2 ambar y 2 verdes), 6 resistencias de 1 KOhm y 2 pulsadores. También una placa de circuito impreso (de 5 cm x 5 cm y con agujeros estañados). El conector cuesta unas 100 ptas. un led 10 ptas. (los hay de alta luminiscencia, brillan más con la misma corriente, y valen unas 30 ptas.), y las resistencias una pta. cada una (Material que fácilmente se encuentra en cualquier tienda de componentes electrónicos)



fig.3

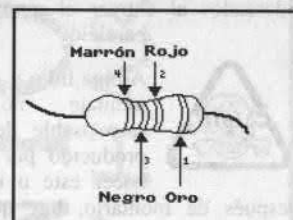


FIGURA 7 (Izqda.) FIGURA 8 (Derecha)

salida se ignora. Por ejemplo, los podemos poner a cero.

Haremos:

MOVE.b #00011110, \$bfe101 ;DRB

Finalmente, para leer el estado de los pulsadores, debemos leer el DRB y testear el bit correspondiente (Supongamos que asignamos un botón rojo al bit 6 y un botón amarillo al bit 7).

Para testear el botón rojo, se hace:

BTST #6, \$bfe101

Para testear el amarillo se hace:

BTST #7, \$bfe101

Notar que para testear el botón del ratón se hace

BTST #6, \$bfe001

y para el del joystick

BTST #7, \$bfe001

Estos bits están a cero si se cierra el pulsador (conexión a masa).

Si quisiera poner un Led en el Joystick y no necesitara el botón, podría poner la línea como salida haciendo:

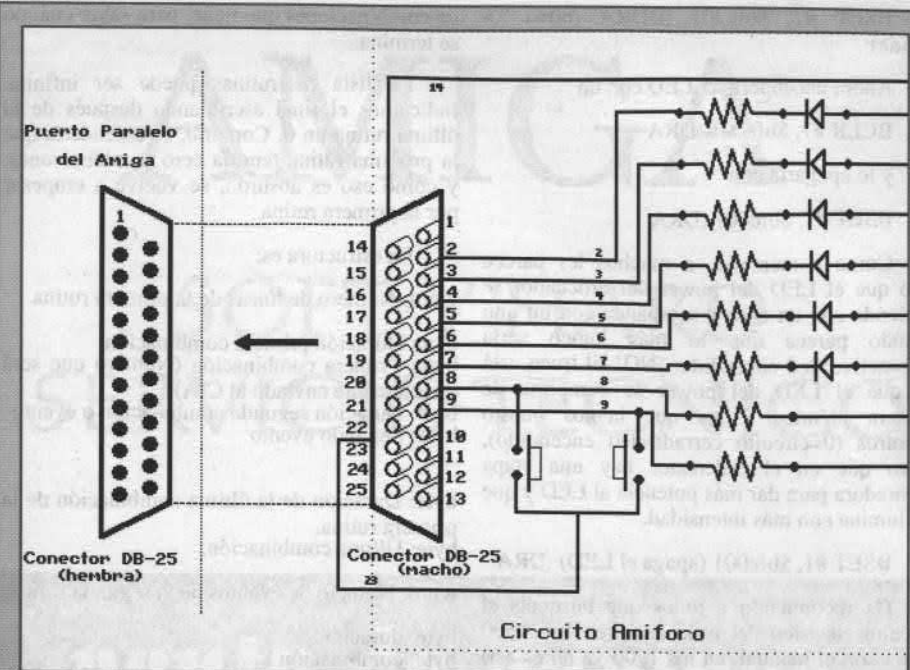
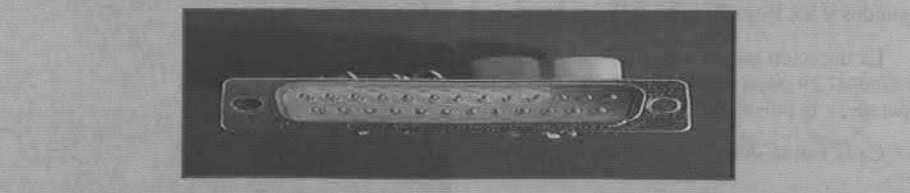
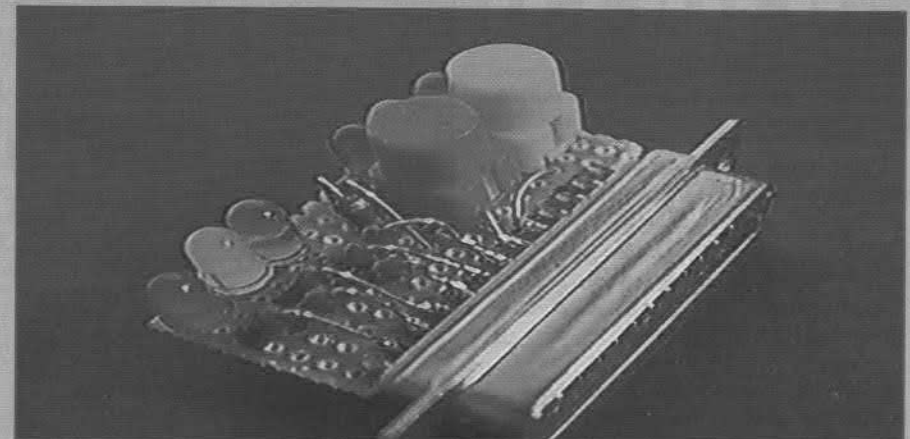
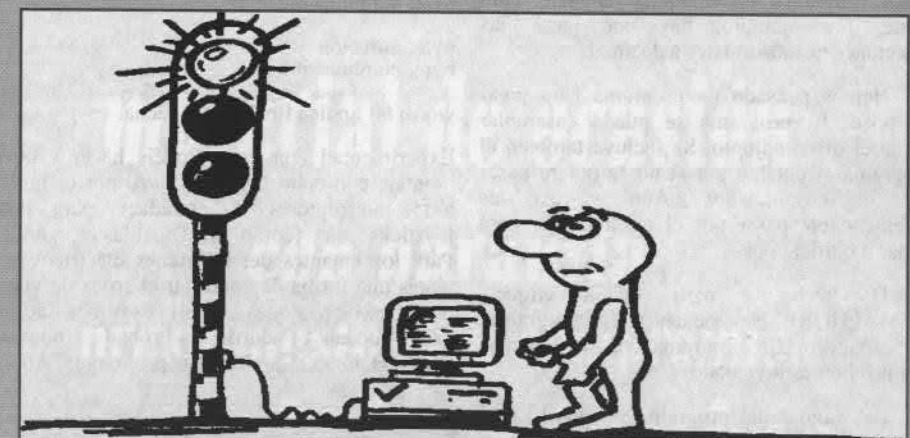


FIGURA 9 (arriba). Esquema de conexión del circuito Amiforo. Observar que únicamente utilizamos 10 líneas de conexión al Amiga (2, 9, 23 y 14). Hay que prestar especial atención al conectar esas diez líneas.

FIGURAS 10, 11 y 12 (abajo). Diferentes vistas del Amiforo ya montado y probado.



BSET #7, \$bfe201 ;DDRA (bit=1 -> Salida)

Ahora encenderá el LED con un

BCLR #7, \$bfe001 ;DRA

y lo apagará con

BSET #7, \$bfe001 ;DRA

Como comentario: a muchos les parece raro que el LED del power del ordenador se encienda con un cero y se apague con un uno cuando parece que lo más lógico sería uno=activo=led encendido. ¡NO! El truco está en que el LED del power se comporta de manera idéntica a los que hemos puesto nosotros (0=circuito cerrado=led encendido), salvo que en el ordenador hay una etapa separadora para dar más potencia al LED y que se ilumine con más intensidad.

BSET #1, \$bfe001 (apaga el LED) ;DRA

Os recomiendo a todos que busquéis el circuito eléctrico del ordenador (en los A500 venía con el manual, en los 1200 ya no es así) y observéis los CIA's (Chips llamados 8520) y sus conexiones. Se puede ver su conexión directa al puerto paralelo y al de juegos.

Antes de llegar al final del artículo, faltaría comentar que aunque nuestros ejemplos han sido en código máquina, se puede programar en cualquier lenguaje. Para Basic, por ejemplo, hay que pasar las posiciones hexadecimales a decimal.

Hemos pensado un programa para jugar un poco. El programa se puede ensamblar desde el disco adjunto. Se incluye también el programa ejecutable por si no se quiere pasar por el ensamblador. Aun y así, es recomendable pasar por el ensamblador para poder modificar cosas.

De hecho, a partir de la etiqueta .SEMAPHORE se encuentra el "programa" del semáforo. Un programa con un lenguaje que nos hemos inventado:

Las salidas del programa son los LED's. Las entradas son los botones de la placa.

El programa se estructura así:

Hay varias rutinas que el semáforo puede ejecutar. Se cambia de rutina con el pulsador rojo.

Cada rutina consta de una sucesión de combinaciones luminosas. Por ejemplo:

Verde1, Rojo2
Ambar1, Rojo2
Rojo1, Rojo2
Rojo1, Verde2
Rojo1, Ambar2
Rojo1, Rojo2

donde cada combinación puede durar más o menos. Por ejemplo, los verdes durarán por ejemplo siete segundos, mientras los ámbar 2 segundos y los Rojo-Rojo tan sólo un segundo.

La duración puede especificarse como "-1 Segundos" en cuyo caso se entenderá que debe esperarse a la pulsación del botón amarillo.

Cada rutina debe ir precedida del número

de combinaciones que tiene, para saber cuándo se termina.

La lista de rutinas puede ser infinita. Indicamos el final escribiendo después de la última rutina un 0. Con ello, se entenderá que la próxima rutina tendrá cero combinaciones, y como eso es absurdo, se vuelve a empezar por la primera rutina.

La estructura es:

word: Número de líneas de la primera rutina

byte: Duración primera combinación

byte: Primera combinación (Número que será directamente enviado al CIA)

byte: Duración segunda combinación o evento

byte: Segundo evento

...

byte: Duración de la última combinación de la primera rutina.

byte: Última combinación.

word: Número de eventos de la segunda rutina

byte: duración

byte: combinación

...

byte: duración

byte: combinación

...

word: Número de eventos de la última rutina

byte: duración

byte: combinación

...

byte: duración

byte: combinación

word: #0 ;indica final de la cadena.

Experimentad con esto. Podéis haceros otro montaje con ocho LED's. Poned por ejemplo todo interruptores (8 entradas), para dos joysticks más (como en Dinablast, ¿no?). Para los amantes del los trenes eléctricos ya téneis una forma de controlar el cruce de vías, o el paso de peatones... o mil cosas más que se os puedan ocurrir. Probad nuevas combinaciones. Cambiad los programas. Aquí,

en la redacción hay quien ha simulado las luces del coche fantástico...(!Sólo le faltaba el coche!)

Si queréis que los eventos no se midan en pasos de un segundo, haced un cambio en MAX EQU 50 y poned otro valor (un valor menor es más rápido).

Los programas de ejemplo que van incluidos en el disco del mes son:

□ SEMAFORO 1: Semáforo normal, en el que los dos semáforos son de coches y con el botón (simulando el centro de control de tráfico) ponemos los semáforos en espera. (2 programas)

□ SEMAFORO 2: Semáforo de coches uno, y semáforo de peatones el otro (el Led ámbar no se usa). Dos programas: Uno simulando el funcionamiento normal, y otro simulando el funcionamiento en el que el peatón debe pulsar su botón (bit #7) para poder cruzar.

Bye y... ¡a respetar los semáforos!

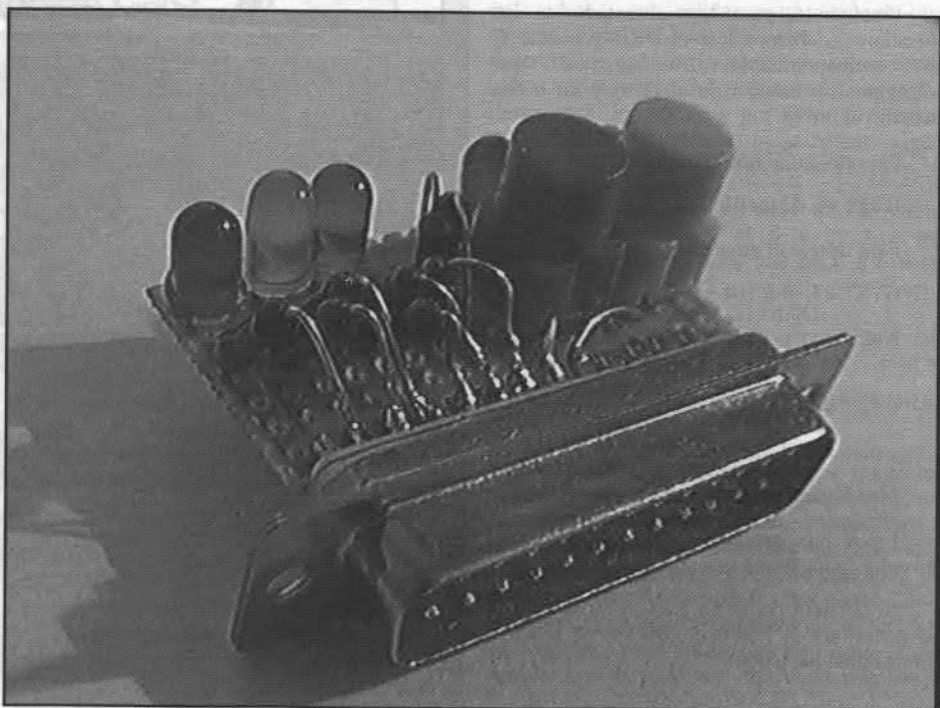
Comentario: Este montaje consume tan poca corriente que se puede conectar y desconectar con el ordenador en marcha, sin embargo, NO ES NADA RECOMENDABLE HACERLO, puesto que se podrían provocar cortocircuitos indeseados al extraer el montaje del puerto paralelo.



Amiga.InFo y el autor de este montaje no se hace responsable de ningún daño producido por negligencia al hacer este u otros montajes.

Después de montarlo, hay que revisar con atención todos los conexiones y evitar errores en las soldaduras. Sólo cuando estemos totalmente seguros, es cuando hay que probarlo con el Amiga.

✓ Observación: NUNCA conectéis al mismo cable 5v y masa NI POR EQUIVOCACIÓN. PUEDE DAÑAR EL ORDENADOR. ●



MACINTOSH
SOFT Y HARD

CIMEX
INFORMATICA

estamos en

Sepúlveda, 167
08011 Barcelona

nos puedes llamar al

tel. 454 70 42

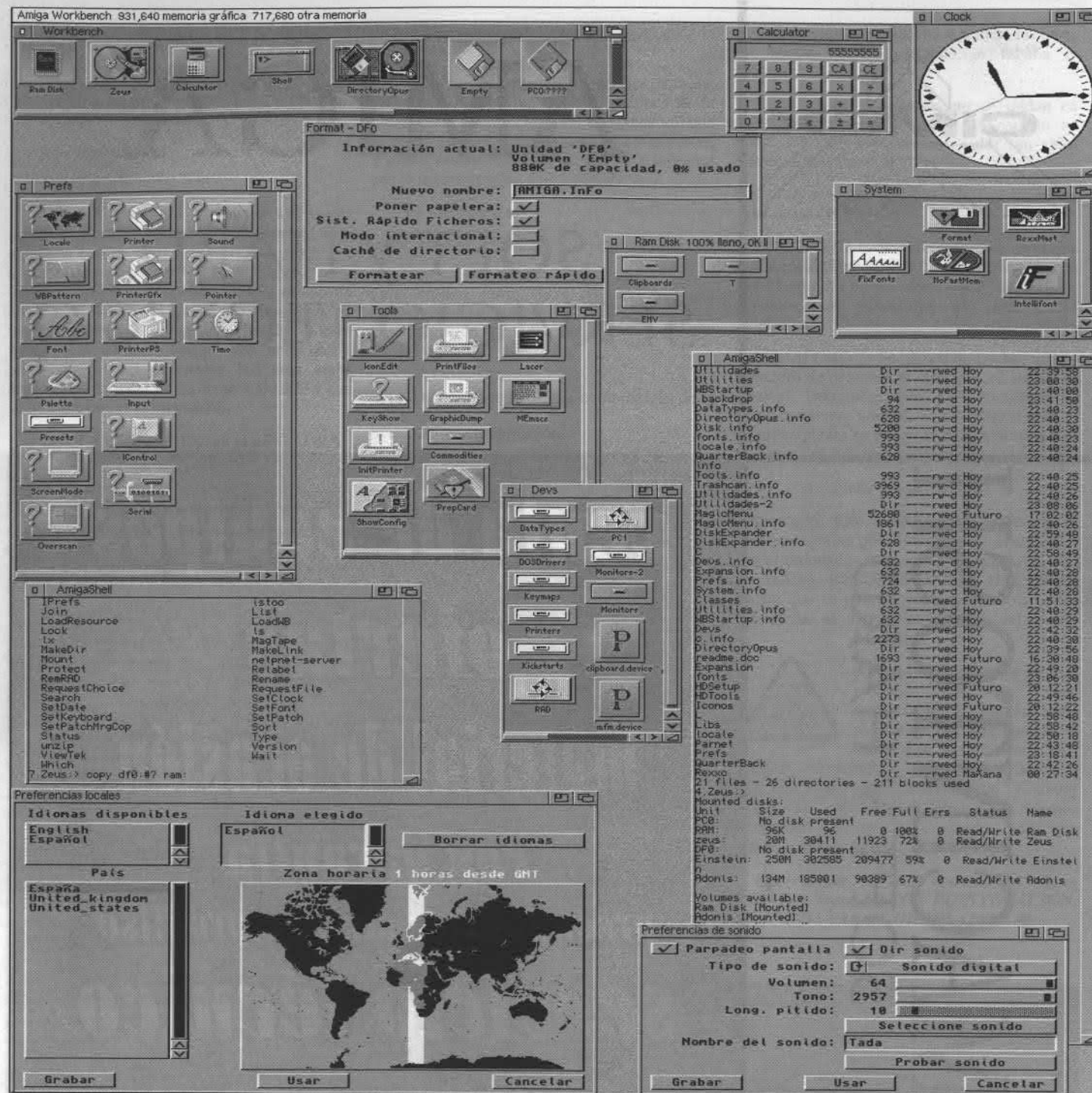
AMIGA

SOFT Y HARD SERVICIO TECNICO

venta de equipos
ofertas
material de ocasión
con garantía
presupuestos sin compromiso
servicio técnico

PC compatibles

SOFT Y HARD SERVICIO TECNICO



INTRODUCCIÓN AL SISTEMA OPERATIVO



Hoy en día todos hemos oído hablar de los sistemas operativos como parte fundamental de un ordenador. Sin embargo, cuando nos enfrentamos al sistema operativo del Amiga, podemos observar que es difícil determinar dónde comienza y acaba, cómo está organizado, qué funciones tiene...y sobre todo, ¿cómo podemos programar nuestras aplicaciones o simplemente ejecutarlas a través (y con ayuda) del sistema? F. Martín nos introduce a una de las más apasionantes y geniales 'piezas de software' programadas para un ordenador:

AmigaDOS (Sistema Operativo Amiga)

Para lograr este objetivo, mostraremos temas relacionados con el manejo y función de cada una (y casi siempre en combinación con otras) de las partes que componen todo el sistema operativo. Debido a que todos los ejemplos que se publicarán estarán escritos en C, los artículos quedan unidos estrechamente al curso del lenguaje C que nuestro amigo A. Moreno realiza en su sección de Amiga.InFo. Las dos secciones se escriben en paralelo y se relacionan para intentar que tanto el lenguaje C como el S.Operativo sean dos herramientas fundamentales para el programador de aplicaciones en Amiga. Así pues, a la hora de escribir un ejemplo en C que ilustre algún punto concreto, las cuestiones propias de este lenguaje se dirigen a la sección de C. Aquí no se comentará propiamente el lenguaje, sino la programación y uso del S. Operativo a través de él.

UN POCO DE HISTORIA

Ocurrió en los años 80. Un grupo de ingenieros (por no decir dos o tres) tuvieron la idea de diseñar una máquina que no siguiese los esquemas de la época. Desarrollaron un hardware avanzado que permitía integrar imagen, sonido y programas en una misma máquina. Simplemente era una caja negra repleta de circuitos, pero que mostraba unas capacidades gráficas y sonoras fuera de lo común. El éxito como máquina de videojuegos estaba asegurado.

Pero no era suficiente. Se podía investigar más y pasar de consola a ordenador. Y así lo hicieron. Nació el Amiga.

Todo el mundo comentaba sus excelentes gráficos y se sorprendían con la calidad del sonido, pero pocos observaban un aspecto que escondía un gran potencial: El sistema operativo.

En aquella época desarrollaron e implementaron ideas en el Amiga, que ahora, en 1994, están adaptando otros sistemas que se consideran modernos. Está basado en un entorno gráfico con diferentes pantallas y ventanas dimensionables, que coexisten con iconos, cajones de ficheros, y otras herramientas gráficas muy similares externamente a sistemas como X Window (Unix) o OS/2 (IBM PC). La utilización de

esta organización gráfica, unida al ratón, ofrece una facilidad inusual para realizar todo tipo de tareas y conseguir que cualquier usuario (de forma intuitiva) aprenda a desenvolverse en las más complejas. Este entorno está potenciado con la posibilidad de ejecutar simultáneamente todo tipo de programas que emplean y comparten todos los recursos del sistema.

Un usuario puede ejecutar al mismo tiempo varios programas e interactuar con ellos para obtener una productividad mayor. Por ejemplo, un programa de dibujo, uno de autoedición, uno de digitalización y un procesador de texto. Todos pueden estar ejecutándose de forma simultánea. Transferir los resultados de una foto digitalizada al programa de dibujo, retocarla a voluntad, añadir un texto que la acompañe y unir todo con el programa de autoedición para imprimir el resultado final, es posible hacerlo con el simple movimiento por pantalla del puntero del ratón y ¡todo esto mientras se escucha un módulo de música de Amiga!

Evidentemente, esta capacidad multitarea no es posible sin el apoyo de un 'hardware' altamente especializado. Para ello, Amiga emplea varios 'custom chips' que incorporan, entre otros circuitos, coprocesadores para gráficos y sonido. (En la sección 'Amiga Interno' encontraremos más información sobre el hardware del Amiga).

CONCEPTOS PREVIOS

En este primer acercamiento al Sistema Operativo, al que llamaremos S.O para abreviar, hay que introducir algunos conceptos que para algunos lectores resultarán desconocidos y, para otros, familiares, pero que al fin y al cabo siempre es conveniente recordar. Las tareas, procesos, dispositivos, memoria y un largo etcétera van a ser muy pronto parte de nuestro lenguaje de 'andar por casa', ya que son necesarios para comprender cualquier tema relacionado con el S.O. Pero no esperemos más, y entremos en materia comenzando por una definición del propio S.O:

□ **S.O:** Un sistema operativo es un programa que actúa como interfaz entre la

máquina y el usuario, facilitando a este último el acceso y ejecución de otros programas. Este interfaz está estrechamente ligado con el hardware de la máquina, y nos proporciona los recursos necesarios para poder realizar las tareas tan comunes como acceso a los discos, memoria, pantalla, impresora, etc. El S.O no es un programa en concreto sino que, generalmente, se trata de diversos programas que unidos forman un entorno de trabajo para el usuario.

□ **Programa y Proceso:** Los programas que ejecutamos, es decir, el conjunto de instrucciones que forman el código y los datos necesarios, son, desde el punto de vista del S.O, procesos. Para nosotros un programa se compone de un código y unos datos, pero para el S.O no es suficiente tener esa información. Necesita también otro tipo de datos para poder integrar el programa dentro del sistema, y ejecutarlo. Información, como pueda ser la zona de memoria donde se carga, ficheros que maneja, recursos que utiliza y otros aspectos que puedan aparecer a la hora de ejecutar el programa, son indispensables y están contenidos internamente en los procesos. Normalmente, esta información adicional es añadida automáticamente a nuestro programa por el compilador o ensamblador que utilizamos. Para el usuario esto es invisible y transparente, puesto que sólo observa como su programa se ejecuta y se le presenta en pantalla con tan sólo un par de toques en el ratón. Sin embargo, para el S.O y el programador, ese programa se convierte en un proceso con todo lo que conlleva.

□ **Tarea:** Los procesos, internamente, tienen asociadas otras entidades ejecutables más primitivas, de nivel más bajo, denominadas tareas. Todo proceso contiene una tarea, además de la información adicional que hemos comentado antes, que permite principalmente el acceso al sistema de ficheros. La tarea está compuesta internamente por un conjunto de instrucciones a ejecutar (el que forma nuestro programa) y sus datos, así como una zona de memoria reservada, donde se almacenan datos temporales, denominada pila. Una de las razones de trabajar con procesos y tareas, es la de diferenciar qué parte de un proceso es la que va a ser ejecutada por el procesador, es decir la tarea, y que parte es simplemente de datos. Es más, pueden existir

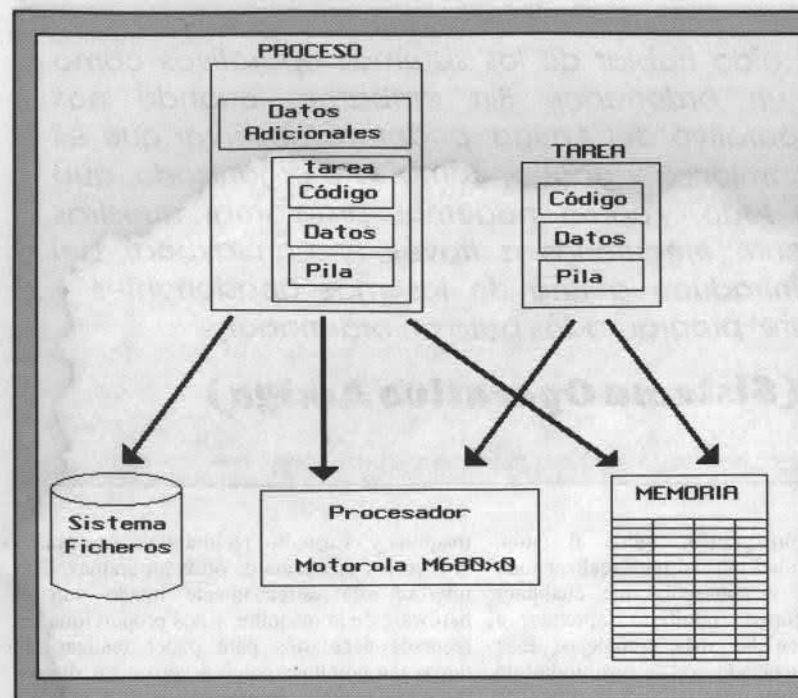


FIGURA 1. Desde el punto de vista del S.O. nuestros programas se convierten en procesos o tareas. ¿Cuál es la diferencia?

- La tarea posee la información mínima para que el procesador pueda ejecutar un conjunto de instrucciones, acceder a unos datos, y utilizar una zona de memoria temporal para datos auxiliares (pila).
- En el momento en que una tarea necesita acceder al sistema encargado de manejar los ficheros, se convierte en proceso. A ella se le une, a parte del código, datos y pila, otra información adicional que es utilizada por el sistema para controlar el acceso a los sistemas de almacenamiento.

¿Por qué diferenciar tarea y proceso?

La tarea es mucho más simple de manejar que un proceso y ocupa menos memoria debido a que no tiene información adicional sobre los ficheros, al contrario que el proceso. El procesador M680X0 ha de dedicarse a cambiar constantemente y de forma rápida la ejecución de las múltiples tareas, y cuantos menos datos maneje cada tarea, más sencillos y eficientes resultarán estos cambios (recordemos que la multitarea está basada en estos cambios rápidos). Por lo tanto, desde el punto de vista del procesador es conveniente utilizar las tareas, y de hecho, no existe nada más que eso para él. Los procesos únicamente existen para el módulo llamado AmigaDOS (ver figura 3).

tareas que se ejecuten en el procesador y formen parte de procesos, o tareas independientes que no estén asociadas a ningún proceso. Estas últimas, por ejemplo, son el caso de rutinas sencillas que no tienen necesidad de acceder a ficheros. La figura 1 nos puede ayudar a ilustrar el concepto de tarea y proceso.

□ **Multitarea:** El S.O. de Amiga es multitarea o multiprogramado, es decir, tiene la capacidad para controlar la ejecución simultánea de múltiples e independientes tareas. El usuario normal tiene la visión de poder trabajar con varios programas a la vez. Para el S.O. y el programador, esta visión es bastante diferente: el procesador es compartido en el tiempo por las tareas, de forma que cada una tiene su propio procesador virtual. En un determinado instante, únicamente una tarea está ejecutándose en el procesador. Cuando transcurra un tiempo, esta tarea deja de ejecutarse y entra otra. De esta forma cíclica, cada tarea toma un tiempo de procesador, lo deja y entra la siguiente. Evidentemente, el acto de cambiar de una tarea a otra es rapidísimo (de 10 a 100 milisegundos), tanto que como resultado final, el usuario tiene la ilusión de que todos sus programas están corriendo simultáneamente. Pero... ¿quién controla el cambio de tareas? La respuesta es simple: El S.O. Pero éste, que es un tema tan apasionante como extenso, merece un próximo artículo.

□ **Procesador:** Es el componente físico* que ejecuta el código de las tareas. Tiene acceso a todo el sistema hardware, recibe y atiende a las interrupciones, controla la comunicación física con otros componentes, el acceso a los buses y muchas más cosas que hacen que sea el eje central de todas las actividades que se realizan dentro del ordenador. Para que nos sea más familiar, también lo nombraremos 'M680X0', ya que en Amiga únicamente se han implementado procesadores de la familia Motorola 68000 (Para más información, ver sección Amiga

Interno).

□ **Memoria:** Es el componente físico que almacena los programas y los datos. Está dividida en casillas (posiciones dentro de la memoria) donde se almacena individualmente cada dato de tamaño 8 bits. Cada casilla tiene una dirección única (valor numérico) que la identifica. Si queremos almacenar un dato, se le envía a la memoria la dirección destino, el valor del dato y la orden de escribir. Si hay que leer, se le envía la dirección de la casilla y la orden de lectura, después la propia memoria nos dará el dato que contiene esa casilla.

El S.O. está basado en direcciones relativas de memoria, es decir, las funciones y estructuras de datos, así como los programas, pueden residir en cualquier parte de la memoria y no necesariamente en una posición única y fija. Incluso pueden reorganizarse dinámicamente dentro de ella según las necesidades y circunstancias. Como resultado, cualquier programa que necesite un bloque de memoria, debe solicitarla al sistema operativo, y este se encargará de buscar una zona libre. Normalmente, los atributos de la memoria son determinados por el tipo de memoria que el programa necesite. Efectivamente, en Amiga existen diferentes conceptos de memoria, los principales son:

Memoria CHIP: Esta memoria es la única direccionable por los 'custom chips' (comentados en la sección Amiga). Este tipo es necesitado cuando existen datos que han de ser accedidos por los canales DMA* de estos chips especiales. Generalmente son datos de gráficos y sonido, así como cualquier dato relacionado con el manejo de los coprocesadores (programas de 'copper*', por ejemplo), aunque también se incluyen los 'buffers' de disco y los 'sprites'.

Memoria FAST: Es la memoria a la cual únicamente puede acceder el procesador. En este caso, los 'custom chips' no tienen acceso a ella, quedando totalmente libre para almacenar

código y datos de los programas. El término 'FAST' (rápida) se debe a la arquitectura de los buses del Amiga. Mientras que la memoria CHIP puede ser accedida por el procesador y los custom chips, con prioridad para estos últimos en caso de acceso simultáneo, en la memoria FAST no se produce esta 'colisión', permitiendo al procesador MC680X0 funcionar con la máxima eficiencia. Evidentemente, un Amiga sin memoria FAST únicamente puede disponer de la CHIP, que se comparte tanto para programas como para los datos de los 'custom chips'. Por este motivo, hay que notar la desventaja que en cuestión de eficiencia se obtiene entre un mismo Amiga ampliado con FAST y otro sin ella (diferencia que se acentúa aún más cuando se utilizan DRAM's* de 16 bits frente a los de 32 bits).

El sistema mantiene una lista de los bloques de memoria utilizados y los libres, siendo posible añadir nuevos bloques a esta lista de forma dinámica*, ya que se reconoce automáticamente la cantidad de memoria física que el sistema posee.

□ **Dispositivo E/S:** Una parte importante del sistema son los dispositivos o 'devices' que permiten acceder al hardware del Amiga. Los dispositivos permiten al programador operar, por ejemplo, con la impresora, un modem, el motor de la unidad de disco, temporizar un suceso, emitir un sonido, etc.. Un dispositivo es un módulo dentro de la jerarquía del sistema operativo que actúa como interfaz. Acepta órdenes y datos y realiza operaciones de entrada/salida basándose en los comandos que recibe. En la mayoría de los casos, interacciona con el hardware interno o externo y tiene una tarea ejecutable asociada que es capaz de procesar comandos mientras nuestras aplicaciones quedan libres para realizar otros trabajos. No son parte del hardware, es decir, son código y datos de control, y por tanto, susceptibles de modificación por parte del programador.

Desde el punto de vista del usuario, los

dispositivos se muestran como simples nombres que van ligados a periféricos, por ejemplo, la unidad de disco (DF0:), disco duro (DH0:), impresora (PRT:), puerto serie (SER:), etc. (Muy a menudo, a estos y otros componentes, los llamaremos en general 'recursos físicos'). El S.O. reconoce automáticamente algunos de ellos y los activa al arrancar el sistema.

Para el S.O. y el programador, los dispositivos son tareas independientes que están a la espera de recibir cualquier comando para activarse y pasar a ejecutarlo, o a la espera de que un suceso ocurra y comunicarlo al S.O. La filosofía empleada en los dispositivos es que las operaciones de E/S deben ser efectuadas de forma consistente, uniforme y estándar. Se puede imprimir un fichero de la misma forma que se puede emitir un sonido digitalizado, tan sólo con mandar a cada dispositivo correspondiente (el paralelo y el de audio) un comando WRITE (escribir) y la dirección del buffer donde se almacenan los datos. El resultado es que el interfaz presentado al programador es esencialmente independiente del recurso físico a controlar y accesible desde cualquier lenguaje de programación. Esta característica expande de forma increíble las posibilidades que el Amiga provee al programador y, finalmente, al usuario.

Los dispositivos soportan dos tipos de comandos: Los estándar, como READ (leer) o WRITE (escribir), y los específicos del recurso físico a controlar, como por ejemplo MOTOR, que activa el motor de la unidad de disco (trackdisk.device), o el comando

READMATRIX, que retorna el estado de cada tecla del teclado (keyboard.device).

Esta estructura de dispositivos es compartida por otros sistemas operativos modernos, puesto que proporciona al sistema modularidad y acceso sencillo, así como una organización eficiente de los recursos disponibles. Es posible añadir dispositivos de forma dinámica, característica poco común en los sistemas operativos como Unix, donde para añadir un dispositivo nuevo hay que recompilar todo el sistema!

□ **Sistema de ficheros:** Es la parte que se encarga de gestionar la información que se puede crear, leer, escribir, borrar, etc. en los dispositivos de E/S. Esta información puede ser ficheros, directorios, protecciones, fechas de grabación, atributos (ejecutable, borrrable, escritura, etc.). Mantiene un control de acceso a ellos, y maneja los errores que puedan existir.

Llegados a este punto, ya tenemos una idea, más o menos clara, de algunos de los conceptos que a partir de ahora aparecerán muy a menudo en esta sección. Sin embargo, estos conceptos por sí solos no representan gran cosa, hay que unirlos para poder tener una visión global del sistema al cual nos enfrentamos.

GESTIÓN, CONTROL, PLANIFICACIÓN

Si reflexionamos un poco, nos daremos cuenta inmediatamente de que la vida se nos haría insostenible sin ciertas cosas tan necesarias como la electricidad, luz, agua, aire

y como no, nuestro Amiga. Pero sería más que insostenible si cuando conectáramos el ordenador tuviéramos que teclear los programas que queremos que ejecute, activar los periféricos uno a uno para poder leer o escribir en ellos, manejar la memoria y tener cuidado que no se llene, controlar donde está situado el ratón y el cursor en pantalla... ¿Quién tendría entonces un ordenador? Algún masoquista. Afortunadamente, esta no es la situación, ya que las anteriores y muchas más tareas las realiza automáticamente el propio ordenador, y para nosotros, sus usuarios -y a veces esclavos- la vida se nos hace menos pesada.

No es muy difícil averiguar que el S.O. es el encargado de controlar, gestionar y planificar el 90 % de todo lo que sucede en un ordenador. El restante 10% queda para nosotros, los usuarios y programadores.

* ¿Qué controla?

Prácticamente controla las actividades de todos los componentes internos o externos electrónicos. Su control es absoluto, puesto que ha de proporcionar una estabilidad y seguridad a todo el sistema.

* ¿Qué gestiona?

Maneja toda la comunicación y movimiento de la información entre los diferentes módulos que lo componen. Avisa y es avisado de los acontecimientos que suceden en su entorno.

* ¿Qué planifica?

¿QUÉ DISPOSITIVOS TIENE EL S.O. DE AMIGA?

Los dispositivos estándar del sistema se pueden clasificar en dispositivos de bajo nivel, encargados de recoger la información directamente sin procesar, y los de alto nivel, que procesan la que proviene de los de bajo nivel. Al inicializar el ordenador o cuando son necesarios, se activan automáticamente los siguientes:

- ◆ **Audio:** Controla los canales de sonido. Puede producir sonidos sintetizados o sampleados, desde un pitido hasta una canción completa.
- ◆ **Clipboard:** Permite almacenar temporalmente los datos que hayan sido cortados, copiados o movidos por diferentes programas. Actúa como un canal de comunicación para intercambiar rápidamente datos entre los programas sin necesidad de grabarlos en disco (Utiliza la memoria RAM como zona temporal de datos).
- ◆ **Console:** Provee el interfaz que interpreta los códigos ASCII y los comandos entrados desde una SHELL.
- ◆ **Gameport:** Controla los dos puertos de joystick/ratón y lee periódicamente los datos que provengan de ellos. Es posible utilizarlo para manejar cualquier otro periférico como por ejemplo un lápiz óptico.
- ◆ **Input:** Combina los datos proporcionados por los dispositivos gameport y keyboard. Permite coordinar y unificar la información que proviene de estos dispositivos para ser utilizada por otros.
- ◆ **Keyboard:** Controla la información entrada por el teclado y la prepara para ser utilizada por otros dispositivos.
- ◆ **Narrator:** Produce el habla sintetizada por los canales de audio.
- ◆ **Parallel:** Controla las comunicaciones por el puerto paralelo y la información a enviar o recibir. Es utilizado normalmente por el dispositivo de impresora, pero también por ejemplo para digitalizadores de imagen o sonido.
- ◆ **Printer:** Convierte un conjunto estándar de códigos de control de impresora (comandos para imprimir texto y gráficos) en otros propios de cada modelo de impresora. Para ello utiliza unas preferencias personalizadas por el usuario que determinan el modelo de impresora, ancho y longitud de página, tipos de letra, etc.
- ◆ **SCSI:** Proporciona los comandos estándar necesarios para la comunicación con dispositivos físicos que sigan la norma SCSI (principalmente discos duros).
- ◆ **Serial:** Controla las comunicaciones del puerto serie.
- ◆ **Timer:** Proporciona funciones de temporización de sucesos y envía interrupciones con periodos de segundos y microsegundos.
- ◆ **Trackdisk:** Controla el acceso directo a las unidades de disco de 3.5" y 5.25".



Organiza, coordina y prevee todas las situaciones que se puedan dar en el Amiga, para poder usar todos los recursos de una forma óptima y eficiente.

Pero veamos realmente por qué el S.O. tiene tanta importancia, a veces un poco infravalorada por no tener conocimiento. Lo haremos con un ejemplo muy sencillo y que hemos realizado miles de veces: conectamos el Amiga, tras unos segundos en los que se activa lo necesario, aparece en nuestra pantalla el WorkBench; desplazamos el ratón hacia el icono del programa que más usamos (¡seguro que la mayoría ha pensado en el Directory Opus!) y pulsamos dos veces el botón izquierdo; tras el instante que dura la carga en memoria, se ejecuta y aparece, en una nueva ventana, el programa. Hasta ahora no ha pasado nada que no conozcamos, por lo menos desde nuestro punto de vista de usuario normal.

Pero imaginemos ahora que pasamos de usuario a superusuario, y que podemos observar cada detalle de la actividad del S.O., desde que encendemos el ordenador hasta que nuestro programa aparece en pantalla.

- 1 - ¡Click! Interruptor ON
- 2 - Se inicializan los registros de los diferentes chips con los valores adecuados.
- 3 - Se verifica y testea la cantidad de memoria disponible.
- 4 - Se comprueba si el 'kickstart' (es una parte

del S.O.) está en ROM* o hay que cargarlo de disco.

5 - Se comprueba si existe un disco en la unidad 0.

6 - Si no hay disco y hay que cargar el kickstart, se solicita un disco de tipo 'kickstart' para cargarlo.

7 - Si no hay disco y el kickstart está en ROM (o se ha cargado ya en memoria RAM), el procesador ejecuta el código que reside en el 'Kickstart', es decir, el S.O. inicia su control del sistema.

8 - Se verifica e inicializa el teclado.

9 - Se comprueba la existencia de expansiones autoconfigurables (memoria, aceleradoras, discos duros...)

10 - Si no se ha introducido un disco, se presenta la pantalla de introducir disco (la típica mano del WB o la animación del disco introduciéndose en la unidad). En caso de no haber detectado la existencia de ningún otro recurso físico que pueda arrancar el sistema (Disco duro, RAD...) se mantiene en espera hasta que se introduzca.

11 - Si existe un disco en la unidad, se comprueba de que tipo es. Si es reconocible por el S.O., se intenta arrancar cargando la primera pista (sector 0 y 1) y pasando el control al programa almacenado en él. Si es un disco no reconocible, por ejemplo, uno no formateado, se vuelve a presentar la pantalla solicitando disco y comienza de nuevo el proceso en 10.

12 - Si el disco es de tipo DOS, se intenta ejecutar el fichero de inicio 'startup-sequence' del directorio 's/'.

Si no existe este fichero, se abre una ventana y aparece una ventana para teclear comandos (Shell).

Aparece en pantalla el cursor y se inicializa el 'sprite 0' para que aparezca el puntero del ratón. A partir de ahora, cada vez que utilicemos el teclado o variemos la posición del ratón, así como si se pulsa algún botón del mismo se ejecutarán unas rutinas. Estas rutinas avisan al S.O. de los sucesos que ocurren, y este actualiza la pantalla según estos avisos.

13 - Si existe el fichero de inicio 'startup-sequence' (ya sea en el disco, o en algún otro recurso físico como el disco duro) se carga y se ejecutan los comandos escritos en él. Se realizan asignaciones de directorios, aparece el disco virtual 'RAM', se montan los restantes dispositivos, y se cargan algunas librerías del directorio 'libs'. También se configuran las preferencias y finalmente aparece la pantalla del Workbench. Se cargan los iconos de los directorios, ficheros, y dispositivos montados correctamente y se muestran en ella.

14 - Se detecta un movimiento del ratón en dirección a un icono, y cuando está situado sobre él, se detectan dos pulsaciones seguidas, del botón izquierdo del ratón. Se intenta cargar el fichero que tiene el mismo nombre que el presentado en pantalla por el icono. En este caso es el 'Directory Opus'. Se reserva un bloque de memoria del tamaño del fichero y se carga del dispositivo al cual pertenece ese icono.

15 - Se ejecuta el código que existe en el fichero y aparece en pantalla el 'Directory Opus'.

Como podéis comprobar, el S.O. tiene que estar atento a todos los sucesos que ocurran, y actuar en consecuencia. Estos 15 son algunos pasos deducidos, ya que si tuviéramos que entrar realmente en hacer un seguimiento exhaustivo, descubriríamos que hay más de 200 puntos que nos hemos olvidado. El S.O. está para ejecutar esos puntos y resumirnos el trabajo de tal forma que únicamente 2 de esos 200 puntos dependen directamente de nosotros: introducir un disco (si no hay disco duro, claro) y pulsar dos veces el icono del programa que queramos ejecutar.

Si generalizamos, es decir, no entramos en detalles, podemos ver que las funciones principales de cualquier sistema operativo son las siguientes (ver figura 2):

- Manejar los procesos (crearlos, destruirlos, bloquearlos...)
- Manejar la memoria (reservar y liberar bloques de memoria...)
- Manejar los dispositivos (leer/escribir información a través de ellos...)
- Manejar los ficheros (crearlos, abrirlos, cerrarlos, borrarlos, actualizarlos...)
- Protección y detección de errores (situaciones de escasa memoria libre, accesos ilegales, gurus...)
- Administración de los recursos del sistema con eficacia y seguridad (coordinar los accesos simultáneos a los dispositivos, compartir la memoria, comprobar las protecciones de los ficheros...)

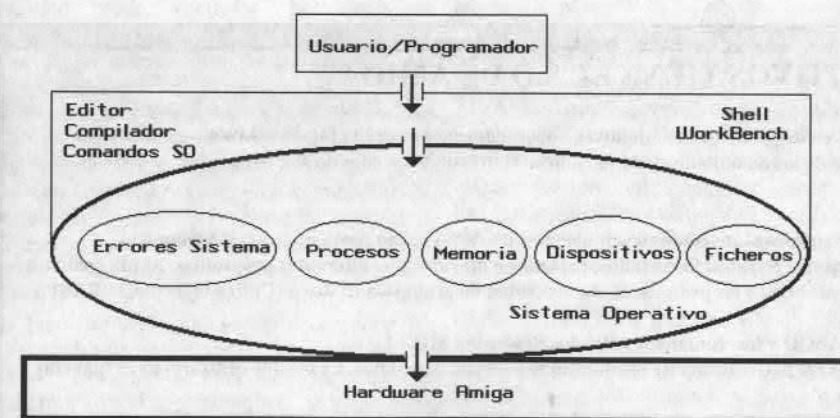


FIGURA 2. Ilustración de las funciones principales de un S.O. Se puede observar como estas funciones de manejo de la memoria, procesos, dispositivos y ficheros están por encima del hardware, es decir, no son parte de él. Así mismo, el usuario normal está por encima del S.O., y "teóricamente" no puede acceder al hardware sin pasar por él.

En la práctica, se salta del nivel de usuario al nivel de hardware muy frecuentemente, sin respetar al S.O. y accediendo directamente a los registros de los chips. Esta forma de acceso siempre ha traído problemas de compatibilidad de código, pero hay que reconocer que se gana velocidad y flexibilidad. Este método es utilizado por los programadores de demos y juegos, pero nosotros seremos 'más legales' y consideraremos al S.O. como un puente entre nuestros programas y los componentes electrónicos del Amiga. Respetaremos las reglas de programación que Commodore aconseja y el acceso directo lo dejamos para la sección 'Amiga Interno'.

Algunos S.O. tienen protegido totalmente el hardware, y no permiten a los usuarios ningún tipo de acceso que no sea contemplado por el sistema. En Amiga no existe esta protección, y cualquiera puede acceder directamente a zonas reservadas y 'colgar' el ordenador.

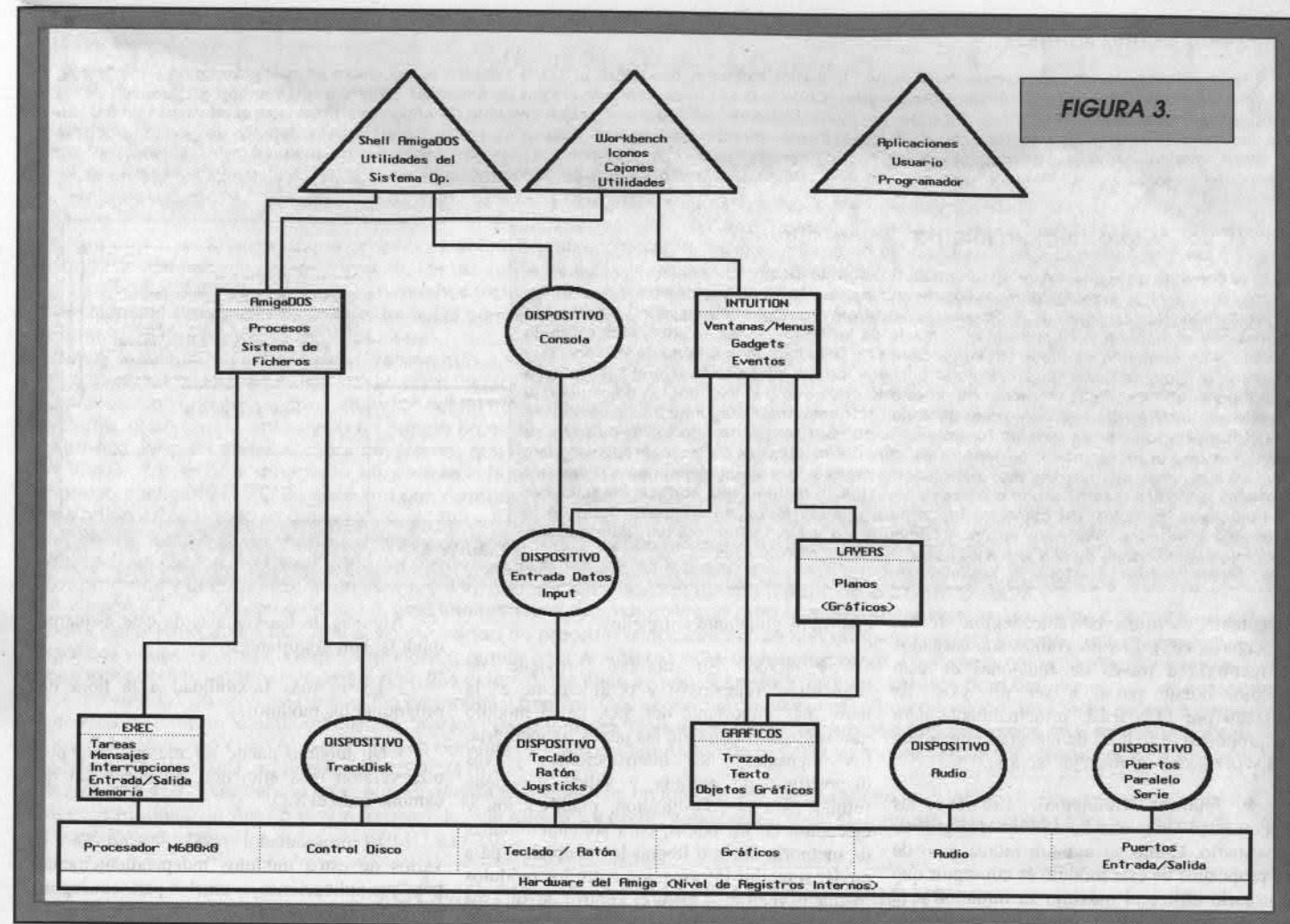


FIGURA 3.

Todas estas funciones fueron incorporadas por los ingenieros al Amiga. Sin embargo, para ellos, existían más, puesto que añadieron ideas clave que tenían como meta conseguir la integración total del S.O. y el hardware. Se propusieron facilitar al máximo cualquier actividad que pudiera realizar el usuario o programador, creando un entorno de trabajo basado en la intuición y la lógica humana. Dividieron el sistema en partes y se les asignó programas de control, es decir, crearon un conjunto de módulos que al unirse formaban lo que conocemos como el S.O.

¿UN S.O. MODULAR?

En Amiga, el S.O. está compuesto por una gran cantidad de módulos formando una jerarquía en función de los trabajos que desempeñan. Estos módulos se comunican y están relacionados entre sí. La información pasa directamente de un módulo a otro utilizando paquetes de datos o 'mensajes'.

Para un usuario normal todos estos módulos son invisibles a excepción de, por ejemplo, los que se encargan de presentar los datos en pantalla o el encargado del acceso a disco. Sin embargo, para el programador de aplicaciones y para el usuario curioso, el conocimiento y la forma de acceder a ellos es indispensable.

Estos módulos se crearon a partir de la necesidad de controlar cada componente

electrónico del Amiga y por otro lado, proporcionar una gran flexibilidad a la hora de desarrollar y adaptar nuevos componentes. El acceso a ellos se realiza mediante llamadas a funciones. Estas funciones son rutinas preprogramadas y residentes en memoria (la mayoría de ellas en ROM). El conjunto de dichas funciones se denomina 'librería' (Ver figura 4 en página siguiente).

A modo de ejemplo de función y librería, podemos imaginarnos que queremos dibujar en la pantalla un círculo. Podríamos programar una rutina que lo dibujase, pero no hay necesidad ninguna, puesto que ya existe una preparada en una librería denominada 'graphics.library'. Con sólo llamar desde nuestro programa a la función 'DrawEllipse()' y pasándole los parámetros correctos, aparecerá en pantalla un círculo. En esta misma librería existe un centenar de funciones dedicadas exclusivamente a gráficos y hay una docena de librerías de todo tipo ¡listas para ser usadas!

Pero a las librerías ya les dedicaremos un artículo completo en otro momento, donde entre otras cosas, descubriremos que los dispositivos son un tipo de librería especial. Ahora veamos que tal se muestran estos módulos en el esquema de la figura 3.

En esta figura, observamos dos elementos diferentes, pero ya conocidos: las librerías y los dispositivos. Los enmarcados en

rectángulos son librerías, y los elípticos son dispositivos. Las líneas verticales representan que módulos se comunican entre ellos. Por ejemplo, el 'Workbench' se comunica con el dispositivo 'Consola', y éste, a su vez, con el de 'Entrada de datos', y sigue hasta que por último se llega al nivel del hardware. La comunicación es bidireccional, es decir, se puede enviar y recibir información de uno o más módulos. Cuando hablamos de comunicar, además de pasar información, hablamos también de la posibilidad de que un módulo utilice las funciones de otro para conseguir un propósito.

¿QUÉ MÓDULOS COMPONEN EL S.O.?

Los dispositivos ya han sido comentados (ver la tabla de dispositivos del Amiga), así que introduciremos brevemente las librerías principales.

□ Librerías:

◆ DOS: Responde a las iniciales de 'Disk Operating System', es decir, sistema operativo de disco. Contiene todas las rutinas relacionadas con el tratamiento de ficheros (abrir, leer, escribir, cerrar ficheros, etc.)

Controla los directorios y toda la información relacionada con el almacenamiento de datos en disco, por ejemplo, los atributos de un fichero

¿Qué es una librería?

En una librería se estructura, ordenan y reúnen todas las funciones que utiliza el S.O. y también se establece un método común y público de acceso, mediante llamadas y parámetros predefinidos. Cada librería puede contener cientos de funciones diferentes que facilitan el desarrollo de los programas enormemente. El programador sabe que puede utilizarlas siempre y en cualquier modelo de Amiga que posea la misma versión de S.O. No ha de preocuparse en programar aspectos de bajo nivel, como puedan ser rutinas de acceso a pantalla, rutinas para la creación de gráficos o acceso a disco, manejo de memoria, control de los puertos de comunicación, creación de ventanas, etc.. La mayoría de estas librerías residen permanentemente en una memoria protegida de sólo lectura (ROM) y otras son cargadas automáticamente de disco a memoria en caso de necesidad.

¿Cuál es la estructura interna de una librería?

Al contrario de lo que se podría suponer, el código de las funciones no está incluido en la propia librería. En su lugar, únicamente existe una instrucción de salto a una dirección donde sí reside la función. La figura de la derecha nos ayudará a comprender mejor la situación. Las partes que componen la librería son el conjunto de las instrucciones de salto, el nodo de la librería, y un segmento de datos. Las instrucciones de salto forman una tabla de vectores que apuntan al inicio de cada función. El nodo contiene toda la información que el S.O necesita para manejar la librería. El segmento de datos está dedicado a la información adicional que pueda necesitarse (Normalmente no es utilizado). Nos podríamos preguntar cuál es el motivo de utilizar esta estructura y no otra: La razón principal está en que usando tablas de saltos se consigue que el programador no tenga que conocer ninguna dirección absoluta, objetivo que los ingenieros de Amiga se propusieron como meta, y que consiguieron como podremos observar. Todo el S.O está basado en direcciones relativas de memoria, es decir, las funciones y estructuras de datos, así como los programas, pueden residir en cualquier parte de la memoria y no necesariamente en una posición única y fija. Incluso pueden reorganizarse dinámicamente dentro de ella según las necesidades y circunstancias.

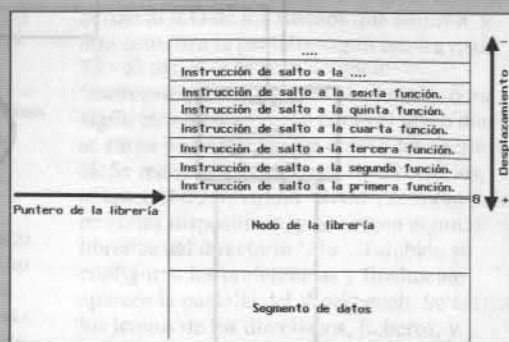


FIGURA 4.

control de colisiones entre ellos.

◆ **Exec:** Su nombre proviene de 'Executive' (ejecutivo) y posiblemente es la parte más importante del S.O. Es el módulo que controla totalmente las tareas, la memoria, los mensajes, las interrupciones y los dispositivos de entrada y salida. Contiene rutinas para la coordinación, planificación y ejecución de las tareas, para solicitar bloques de memoria libres o liberar los ocupados, para enviar o recibir información entre los módulos mediante mensajes, para el control de sucesos que puedan interrumpir al procesador, y para el manejo de las entradas y salidas de datos en los dispositivos. Esta librería es la encargada de manejar la multitarea y es utilizada principalmente por el módulo AmigaDOS.

Todas estas librerías son las más importantes dentro del S.O, pero no son las únicas. Existen otras 'secundarias' que proporcionan un gran número de funciones de todo tipo al programador (funciones matemáticas, interfaz de ficheros, manejo de iconos, carga de fuentes de caracteres, etc.). También es posible crear librerías con nuestras propias funciones, y añadirlas al sistema cuando las necesitemos.

¿QUE RELACION EXISTE ENTRE LOS MÓDULOS?

Una de las ideas clave para comprender el funcionamiento de estos módulos es saber como se relacionan entre ellos y por qué motivo lo hacen.

'Divide y vencerás' son las dos palabras con las que podríamos definir la filosofía empleada a la hora de crear las partes del S.O. Cada trabajo es dividido y realizado por un módulo distinto, y cada uno de ellos recibe o envía la información con la que trabaja a otro módulo. Separados controlan cada uno una parte del Amiga, pero unidos consiguen crear una base activa que nos permite controlar la totalidad del sistema.

Algunas de las ventajas de este esquema modular son importantes:

- Existe más flexibilidad a la hora de programar los módulos.

- Un módulo puede ser reemplazado por otra versión más eficiente, sin necesidad de cambiar todo el S.O.

- El programador puede utilizar uno o varios de estos módulos independientemente para sus aplicaciones, o unirlos para conseguir un resultado concreto.

La desventaja principal es que también hay que crear un sistema de comunicación común y fiable entre todos los módulos, y esto implica principalmente pérdida de velocidad del S.O. Sin embargo, en Amiga se ha diseñado un sistema de comunicación rápido y seguro, utilizando puertos y mensajes, que cargan mínimamente al procesador. Este sistema será comentado cuando hablemos detalladamente de la librería 'Exec'. También hay que parametrizar cuidadosamente cada función dentro de las librerías, definiendo qué registros y valores hay que pasar en las llamadas, y los resultados que devuelven.

La idea principal a tener en cuenta es como se relacionan los módulos en algunos de los trabajos propios de un S.O, como puedan ser, por ejemplo, los siguientes:

□ **Presentación en pantalla:** La pantalla, con sus ventanas, menús, gadgets y demás herramientas gráficas es uno de los aspectos del S.O que el usuario puede ver directamente. Normalmente, la primera pantalla con la que trabajamos es la del Workbench. ¡Ojo! El Workbench en sí es no es parte del S.O, es un programa de aplicación, que abre una pantalla pública a la que otros programas pueden acceder y utilizar. Permite a los usuarios usar el sistema de ficheros a partir de iconos y ofrece al programador un conjunto de funciones para manejarlos. Está situado en la parte más alta de la jerarquía, y podemos ver en la fig.3 como se comunica con AmigaDOS



¿QUÉ VERSIÓN DEL S.O TENGO?



Es ley de vida, todo evoluciona, y el sistema operativo del Amiga no podía ser menos. Desde su creación en los 80 hasta hoy día, han aparecido diferentes versiones que mejoran a las anteriores. Es conveniente conocer la versión que poseemos y evitar encontrarnos con la sorpresa de que muchos de los ejemplos que saldrán en esta sección no nos funcionen, o haya que modificarlos para que podamos compilarlos. Veamos que versiones existen, y cuales son las que nos interesan:

- ◆ **Versión 1.2:** Es la primera que apareció y actualmente está obsoleta. Nadie que quiera programar o trabajar seriamente con Amiga puede hacerlo con esta versión. Los usuarios que estén en esta situación deben actualizarse urgentemente.
- ◆ **Versión 1.3:** Esta versión mejoró en muchos aspectos a la 1.2, y muchos de nuestros lectores seguro que la tienen. Existen muchísimos programas que son compatibles con ella. Sin embargo, se ha quedado primitiva. Los programas de esta sección podrán correr, pero con bastantes modificaciones. Es muy recomendable instalar una versión más reciente.
- ◆ **Versión 2.0:** El S.O del Amiga dió un paso de gigante con esta versión. Se modificaron profundamente muchos aspectos internos y externos. Con una nueva apariencia 3D mucho más profesional y atractiva, así como la mejora global en velocidad de todo el sistema, el 2.0 será nuestro punto de partida en esta sección. Además, incorpora multitud de funciones nuevas para el programador y usuario. Para ser sinceros, todos tendríamos que tener, como mínimo, esta versión.
- ◆ **Versión 2.1:** Es básicamente la 2.0, pero se le han añadido nuevas capacidades como, por ejemplo, el sistema 'locale', que permite configurar el S.O en el idioma que deseemos, incluyendo el castellano.
- ◆ **Versión 3.0:** Cuando aparecieron los A1200 y A4000 con nuevas capacidades gráficas, se volvió a dar otro salto en el tema de hardware, y el S.O 2.1 se modificó para poder trabajar con los nuevos chips. Todos los programas corren con las versiones 2.0 y 3.0 del sistema, pero están comenzando a salir programas que requieren la 3.0, puesto que incorpora nuevas funciones muy potentes. Los programas que aparezcan en esta sección, correrán sin ningún problema a partir de la 2.0 en adelante.
- ◆ **Versión 3.1:** Es básicamente la 3.0, pero internamente se ha modificado para que pueda controlar unidades de CDROM formando parte del sistema como dispositivo, sin necesidad de programas adicionales. Se han añadido nuevas funciones, nuevos modos gráficos y soporte para las tarjetas gráficas (permite a los Amiga NO AGA correr programas AGA con tarjetas 24 bits, por ejemplo con la Picasso II, y abrir el workbench en 256 colores). También se han optimizado algunas librerías para obtener mayor velocidad en las actividades de gráficos de Intuition (ventanas y demás). Como curiosidad, también se han añadido unas funciones para convertir gráficos de formato 'chunk' (el utilizado por los PCs) a formato 'planar' (el normal de Amiga). Esto no parece muy interesante, pero hay que tenerlo en cuenta, puesto que se consigue emular el chip AKIKO de la CD32 y permite ejecutar los juegos de la consola en cualquier Amiga AGA equipado con CDROM (I).
- ◆ **¿Versión 4.0?** Esta versión es la que se prevee que saldrá en un futuro (esperemos que no muy lejano) cuando aparezcan los nuevos modelos de Amiga que incorporen los tan famosos y comentados chips AAA. Se rumorea que entre la 3.0 y la 4.0 habrá tantas mejoras como las hubo entre la 1.3 y la 2.0.

¿CÓMO PUEDO ACTUALIZAR MI VERSIÓN?

El método de cambiar de versión depende del modelo de Amiga que tengas, pero generalmente suele ser sencillo. Se puede instalar un nuevo S.O de alguna de estas tres formas:

- 1- Extraer la ROM del interior del Amiga, y colocar una nueva. Esta es la opción más eficiente y segura. El chip es fácil de extraer, puesto que está colocado en un zócalo. Utilizando un pequeño destornillador para hacer palanca, el chip saldrá sin ningún problema. Al introducir la nueva versión hemos de fijarnos en que la muesca que tiene el chip en uno de los costados coincida con la que hay en el zócalo del Amiga. Esta opción pueden emplearla los que tengan un A500 o A2000.
- 2- Conseguir los discos especiales de tipo 'kickstart'. Esta opción sirve únicamente para algunos A3000. Al encender por primera vez el ordenador (no después de un reset), pulsar simultáneamente los dos botones del ratón y aparecerá un menú que permitirá cargar e instalar la nueva versión del S.O.
- Es muy difícil encontrar este tipo de discos, y es más recomendable utilizar la opción 3.
- 3- Si tienes un mínimo de 1 mega de memoria ampliada, puedes cargar el nuevo S.O en esa memoria, no importa que modelo de Amiga tengas. Es necesario un programa de arranque (está en muchas librerías de Dominio Público, por ejemplo Fred Fish) y un fichero donde se haya volcado toda la ROM del S.O. Si es la 2.0, ocupará 512kb. Con este método se copia el fichero a memoria cada vez que inicializas el ordenador en frío (se conserva en memoria después de un reset), y es prácticamente igual que si hubieras cambiado la ROM físicamente (como en la opción 1). No sabemos hasta que punto es legal copiar la ROM a disco y en teoría únicamente algunas personas como por ejemplo 'developers' que tengan permiso de Commodore pueden utilizar esta opción (Son personas que desarrollan e investigan nuevos productos de Amiga).

¿DÓNDE PUEDO CONSEGUIR UNA NUEVA VERSIÓN?

Actualmente están disponibles los discos y la ROM de la versión 2.04 del S.O, así que sólo les interesa a usuarios de los A500 o A2000, puesto que todos los restantes modelos ya la incorporan de fábrica (v2.0 en el A500+, A600, A3000 y v3.0 en A1200 y A4000). Es un kit de ampliación disponible en muchos distribuidores de Amiga. Pregunta en el más cercano, o en caso de no encontrarlo, hazlo directamente a CIB en Madrid (Tlf. 91 632 27 65) o Amiga Center en Barcelona (Tlf. 93 276 38 06). Ellos importan este kit, aunque últimamente andan escasos de existencias por los problemas de suministro de Commodore. Está a la venta aproximadamente por unas 8.000 ptas. (según distribuidor).

Recientemente también se ha puesto a la venta la versión 3.1 (ROM, 6 discos y manuales de Workbench, AmigaDOS y ARexx) para todos los Amiga no AGA. En el A1200/A4000 no es necesario el cambio de ROM, únicamente hay que instalar los 6 discos del sistema. No tenemos noticia de que alguna empresa española lo importe, y en el extranjero, hasta el momento, sabemos que lo comercializa la compañía alemana Village Tronic (para más información, contactar con el distribuidor Blittersoft en Reino Unido, tlf. 0908 220196). En el momento en que esté disponible en algún distribuidor nacional lo analizaremos a fondo para vosotros.

e Intuition. El primero lo utiliza para tratar con los procesos y los ficheros que son representados por los iconos, y el segundo crea el entorno de pantallas, ventanas y menus cuando estos mismos procesos son ejecutados.

❑ **Interfaz con el usuario:** El contacto directo con los ficheros lo obtenemos normalmente al abrir una Shell (También llamado CLI o Command Line Interface). El usuario puede escribir y dar ordenes en forma de comandos (dir, list, type...etc). Para introducir estos comandos se utiliza el dispositivo Consola, donde son interpretados los códigos ASCII, es decir, los caracteres que escribimos en la ventana de la Shell y después de ser procesados se envían al módulo AmigaDOS. Este a su vez carga en memoria el fichero del comando e intenta ejecutarlo, tratando los errores en caso de que ocurran (por ejemplo, cuando no existe el comando que hemos escrito). La Shell actúa de intermediaria entre el módulo AmigaDOS y el dispositivo Consola.

❑ **Emisión de sonido:** La producción de sonido es manejado exclusivamente por un único dispositivo, el de de audio. El se encarga de controlar directamente el hardware para conseguir emitir sonidos sintetizados o digitalizados por los cuatro canales.

❑ **Manejo de los puertos de comunicaciones:** También los puertos paralelo y serie son controlados por sus respectivos dispositivos, proporcionando un interfaz entre el usuario y el exterior del ordenador. Podemos controlar la configuración de los puertos y enviar/recibir información a través de ellos.

❑ **Manejo del teclado y ratón:** Cuando pulsamos las teclas del teclado, o movemos el ratón o joystick, el dispositivo correspondiente se pone en marcha recibiendo el código de la tecla pulsada o las coordenadas del puntero

del ratón, por ejemplo. Estos códigos y coordenadas no son procesados todavía, sino que son preparados para ser utilizados por otro dispositivo que combina las dos informaciones para producir lo que se llama eventos o sucesos. Estamos hablando del módulo llamado 'Input' o 'Dispositivo de entrada de datos'. Por ejemplo, algunos de los eventos que procesa son el cierre, cambio de tamaño y profundidad de las ventanas. Cuando nos situamos con el ratón sobre el gadget de cierre de ventana (o cualquier otro gadget) y pulsamos el botón izquierdo, el dispositivo lee la información, en este caso del puerto del ratón, y la envía al otro dispositivo 'Input'. Este a su vez, recoge los datos y avisará a otro módulo de que se ha producido un evento de tipo cerrar ventana, y que actúe en consecuencia y cierre esa determinada ventana (La librería intuition es la encargada de esta tarea).

❑ **Manejo de los ficheros:** El método más directo para observar como el S.O maneja los ficheros es abrir una Shell y escribir comandos en ella. Un simple 'dir' es interpretado por la shell como un comando y es enviado al módulo AmigaDOS para que sea ejecutado como un proceso, e inmediatamente se creará una tarea manejada por el módulo 'Exec'. Cuando esto ocurra, se utilizarán varios dispositivos como el 'trakdisk' o el 'scsi' para leer el directorio de los ficheros contenidos en un disco o disco duro, y se enviará esta información a la shell, para que nos la presente por pantalla.

EN EL PRÓXIMO ARTICULO...

Hasta aquí hemos introducido el S.O. Ahora ya sabemos su forma general, y como se estructura internamente. Tenemos una idea global que nos permitirá en el próximo artículo abordar de una forma exhaustiva y profunda

todo lo relacionado con las librerías, su composición, como se abren y cierran, como podemos construirlas una propia, como las maneja el S.O, etc. Incluiremos ejemplos en C que traten con ellas, y resumiremos las funciones más importantes que cada librería posee.

Avisamos de que para poder compilar nuestros ejemplos, son necesarios un compilador de C (¡como el que hemos incluido en el disco del mes de la revista!) y los ficheros de descripción llamados "Includes" de la versión 2.x o 3.x del sistema operativo. Estos ficheros son propiedad de Commodore, y al no tener licencia para distribuirlos nos ha sido imposible incluirlos en el disco. Para conseguirlos existen varias formas:

1- Comprarse un compilador comercial. Existe la versión 3.0 del DICE a la venta y la versión 6.5 del SAS C. Ambos incluyen todo lo necesario para poder programar de una forma profesional con un entorno multiventana, compilación y edición directa de uno o más fuentes, manejo automático de errores, debugger, ensamblador, documentación muy extensa de todas las funciones y utilidades, etc.

2- Ponerse en contacto con Commodore a través de CATS (ver la sección curso de C para más información).

3- Existe la posibilidad de que consigamos contactar con el autor del compilador de C DICE 3.0 y proponerle que nos envíe una licencia de uso colectiva. Contacta con nosotros si estás interesado en este tema.

Por último, recuerdo que ante cualquier duda, problema, crítica o simplemente comentario que se plantee, os podéis poner en contacto conmigo a través de nuestro APDO. de correos. En cuanto sea posible, publicaré mis direcciones de FIDO e INTERNET para correo electrónico directo y rápido. ●

¿ QUÉ SIGNIFICA...?

- ✓ **Recursos del sistema:** Diremos que un recurso es cualquier medio utilizado por el sistema operativo para realizar una actividad. Pueden ser recursos del sistema una unidad de disco, impresora, memoria, procesador, etc.
- ✓ **Componente físico:** Los componentes físicos son aquellos que existen en el interior del ordenador o periférico. Son ejemplo de ellos cualquier chip o circuito electrónico.
- ✓ **Custom Chips:** Son circuitos electrónicos especializados en realizar tareas concretas de una forma muy eficiente y a alta velocidad. Por ejemplo, mover bloques de datos de una zona de memoria a otra, trasladar información entre el ordenador y un periférico, mover objetos gráficos, etc. El tipo de diseño y fabricación de estos chips es el más complicado y caro de los existentes.
- ✓ **Canales DMA:** Cuando un componente físico puede acceder directamente a la memoria ya sea para escribir o leer, se dice que posee un canal DMA, o lo que lo mismo pero en inglés, Direct Memory Access. Este acceso se realiza sin que el procesador intervenga en ello, quedando libre para realizar otras tareas más importantes.
- ✓ **Buffers:** Zona temporal de memoria utilizada para almacenar datos que posteriormente se trasladarán a otro destino.
- ✓ **Sprites:** Objetos gráficos que aparecen en la pantalla y que poseen una gran movilidad, utilizando para ello un chip especializado y canales DMA. El puntero del ratón es un ejemplo de 'Sprite'.
- ✓ **DRAM's:** Tipo de memoria utilizada en los ordenadores que se caracteriza principalmente por tener una forma de acceso directo y aleatorio, se puede leer/escribir información y perder los datos en caso de no existir un refresco continuo de los mismos. (Dynamic Random Access Memory)
- ✓ **"añadir de forma dinámica":** Esta expresión significa que podemos añadir o destruir el objeto en cuestión (datos, dispositivos, programas, memoria...) en cualquier momento, cuando sean necesario. El Sistema Operativo tiene siempre constancia de cualquier objeto añadido o eliminado.
- ✓ **ROM:** Tipo de memoria al que únicamente se puede acceder para realizar lecturas de su contenido. Es una memoria generalmente fabricada para almacenar datos que no puedan ser modificados o borrados. (Read Only Memory)

¡¡ Ya en España !!

The Communicator Lite
13.900 Ptas.

The Communicator MIDI
20.900 Ptas.

Photolite (PhotoCD para CD32)
8.900 Ptas.

AMIGA 1200 + HD 215 MB
115.900 Ptas.

AMIGA 1200 + HD 260 MB
120.900 Ptas.

AMIGA 4000/030 + HD 215 MB
269.900 Ptas.

AMIGA 4000/030 + HD 260 MB + 68882
274.000 Ptas.



Ordenador Amiga 1200 67.900 Ptas
Ordenador Amiga 4000/030 222.900 Ptas
Amiga CD 32 62.500 Ptas
Monitor 1084S 40.000 Ptas

Impresora EPSON Stylus 800 43.500 Ptas
Impresora EPSON Stylus COLOR .. 99.900 Ptas
Impresora EPSON LQ-150 Color 45.500 Ptas
Scanner EPSON GT-6500 122.900 Ptas

ModemFax ACEEX 14400 42.000 Ptas
Disco Duro SEAGATE 215 MB 39.900 Ptas
Disco Duro SEAGATE 260 MB 45.000 Ptas
VIPER 030 con MMU MKII 44.900 Ptas
M-Tec 68030/ 28/ 4 MB con MMU 59.900 Ptas
A-1200 Memory PC 1202-8 (0 Mb) .. 24.900 Ptas
Kit CD-Rom Mitsumi + TANDEM 50.900 Ptas
CD-Rom Mitsumi interno (x2) 39.900 Ptas
CD-Rom Panasonic SCSI interno... 29.900 Ptas
Control. OKTAGON SCSI-II 24.500 Ptas

Art Department Pro 2.5 35.900 Ptas
Brilliance 27.900 Ptas
Desktop Dinamite Pack 9.700 Ptas

Plaza Pedro IV, 3, local 9
08120 La Llagosta (BARCELONA)
Tel./Fax. (93) 560 76 12

Somos Especialistas en AMIGA



A.C.U.

Amiga Club UPC



Todos los que tenemos un Amiga sabemos que es un ordenador fuera de lo común por sus características y también sabemos que muchos usuarios quieren información, programas, ideas, y aplicaciones concretas que suelen estar fuera del alcance de la mayoría de nosotros... En ACU nos hemos asociado cien personas de la Universidad Politécnica de Cataluña, con la intención de intentar abrir un camino al usuario de Amiga y establecer unas bases para poder introducir al Amiga en áreas de aplicación serias. Nos consta que somos muchos los usuarios de este magnífico ordenador y gracias a ACU nos conoceremos para intercambiar todo tipo de ideas, información, programas, experiencias... en definitiva, cualquier cosa relacionada con Amiga.

Hasta pronto, y no olvides hacerte socio...

SOLO EL AMIGA LO HACE POSIBLE!!! A TRAVES DE ACU!!!

Si queréis más información, poneros en contacto con nosotros por carta a:

A.C.U. AMIGA CLUB U.P.C.
Casa del Estudiante del Módulo B5
C/Gran Capitán s/n
08034 Barcelona

Un aficionado a Bits & Bauds nos cuenta su primer encuentro con el modem...

Os voy a contar una historia verídica, una experiencia personal de cómo y porqué me introduje en este mundo de las BBS y las comunicaciones. No sé si por suerte o por desgracia mía.

La historia se remonta unos tres años atrás. Yo era un usuario de Amiga de lo más típico, tenía un A500 con un mega de RAM, unidad externa, etc. y por supuesto, nunca se me había pasado por la cabeza el usar un modem. Pensaba: "si ya puedo conseguir programas de otra manera, tendré que comprarme otro cacharro y encima gastar teléfono, ¡De ninguna manera! Realmente la ignorancia, algunas veces, hace decir verdaderas sandeces.

En aquel entonces yo sabía, o creía saber, lo que era una BBS, más o menos.

La idea que tenía era la de un ordenador conectado al teléfono permanentemente, a través de un modem, al que yo, con mi ordenador y mi modem, podía llamar y coger los programas que allí estaban. También sabía que era posible enviarse mensajes, pero que eso no era lo principal, las BBS estaban para intercambiar programas por teléfono. Si alguien tenía acceso a mucha variedad de programas, los dejaba en la BBS y si quería alguno en especial o le hacía falta por algo lo buscaba por allí y si lo encontraba podía copiarlo tranquilamente. Otro "problema" que le encontraba a este sistema de intercambio era que todo, absolutamente todo, era dominio público, y por aquel entonces, creía que el dominio público eran programas malos, que no funcionaban bien y bastante inútiles. No entendía nada, ni conocía el dominio público, ni como funcionaba; absolutamente nada.

Pero yo tenía la sana costumbre, de vez en cuando (cuando el dinero alcanzaba), ir a comprarme alguna revista inglesa, para enterarme de qué pasaba, referente al Amiga, fuera de nuestras fronteras. Y en una de ellas, quizás una Amiga Computing, venía un artículo muy completo sobre todo este mundo de las comunicaciones. Al principio no le presté mucho interés, como siempre nos creemos que lo sabemos todo, pero otro día, cuando ya me había leído lo que consideraba más interesante de la revista, cogí otra vez el artículo, y no sé porqué, me entraron ganas de leerlo. El artículo era muy completo y amplio, y no sé, parecía que detrás de todo aquello, había algo más, algo más que mi precaria idea inicial. Quizás por los titulares alentadores: "The world of comms, A new dimension, A enormous window to the rest of the planet", etc.

Así que, como sitiéndome atraído por una extraña fuerza, me puse con un diccionario en una mano y la revista en la otra, a descifrar lo que allí se decía, y poco a poco fui descubriendo cosas nuevas; a medida que avanzaba en el artículo, me iba sorprendiendo cada vez más.

Allí se hablaba de comunicaciones, en el sentido más amplio de la palabra. El objetivo principal era el de comunicarse con otra gente, con otro mundo, no era el quedarse encerrado,

literalmente, en tu ordenador con tus programas y tus cosas, era abrir los ojos y sacar la cabeza a otro mundo, que hasta aquel entonces, era totalmente desconocido para mí. En el artículo se hablaba de personas que charlaban a través del ordenador, gentes que comentaban cosas sobre cualquier tema, gente que se conocía hablando de un procesador de imágenes y que llegaba a conocerse personalmente y ¡se hacían amigos! Realmente lo que allí contaban iba mucho más allá de mi primera y primitiva idea.

Parecía muy interesante, pero enseguida pensé, que Inglaterra, como Estados Unidos, era otro mundo, y que era una lástima que aquí esto no funcionara como allí, y lo que estaba claro es que lo que no iba a hacer era llamar al extranjero! ¡Nuestra querida Telefónica pide un poco más de dinero que si llamas aquí! Así pasó el tiempo, y esta pequeña inquietud, o quizás curiosidad, de qué era ese mundo, permanecía en un rincón de mi cabeza, agazapada y esperando el momento oportuno para salir.

Y un día, un brillante día de primavera y, aparentemente, un día normal de instituto, hablando con un compañero de clase, con el que, de vez en cuando, hablaba del Amiga, (él también tenía un A500) me contó que hacía unas dos semanas se había comprado un modem, y que era una maravilla. Que un amigo le había dicho que existía una BBS exclusivamente de Amiga y que era fantástico, que enviaba mensajes, preguntaba cosas que no sabía, se enteraba de rumores, comentaba con otros usuarios qué programas les gustaban más a ellos y que problemas daban, ... Estaba contándome lo que yo, unos meses atrás había leído en esa revista inglesa, no podía creerlo, era cierto, y me lo estaban contando allí mismo, y este sitio donde llamaba era ¡aquí! ¡en mi ciudad! De verdad que no podía creerlo de ninguna manera, pero lo estaba oyendo de viva voz.

A los pocos días y debido a mi pesadez e impaciencia, fuimos a su casa para poder comprobar con mis propios ojos lo que sucedía y si era tan fantástico como lo contaba. La verdad, me parecía como un sueño que existiese tan cerca nuestro ¡y sin antes habernos dado cuenta! El hecho de acceder a un BBS auténtica, ¡como las inglesas!, me parecía muy irreal.

Usaba un programa especial, algo diferente de los que estaba acostumbrado a ver. Simplemente era un pantalla limpia totalmente, sin ningún carácter escrito y la habitual y normal barra de menú. Accedí a una especie de lista de BBS. Allí me contó que tenía los números de teléfono de varias BBS a las que él llamaba. Eran números de teléfono normales, como si fuese el de un amigo, pero la diferencia es que el que llama ahora es el Amiga y el que contesta es otro Amiga, pero, que aparte de esto no hay ninguna diferencia, ni de precio, ni de nada, ¡tampoco de ruido! Al seleccionar, la que era para él, su BBS habitual y preferida, el modem empezó a marcar un número de teléfono, y al contestar al otro lado de la línea, su modem produjo unos pitidos extraños, y ante mi cara de sorpresa, me comentó que era el protocolo que existe entre

los dos ordenadores: El uno informaba al otro de la velocidad que podía ofrecer, si había compresión de datos, etc. de alguna manera, esas dos cajas con un circuito integrado, hablaban entre ellas y se ponían de acuerdo en una especie de lenguaje, a base de pitidos.

Al entrar salió una carátula de bienvenida a la BBS, y le pidió el nombre y un apellido. Él puso su nombre y apellidos reales, a mi me sorprendió un poco y le pregunté que porqué no se inventaba un nombre y así no ponía el suyo, pero pronto me respondió, contundentemente, argumentando que vaya chorrada había dicho, ¿por qué no podía poner su nombre real? ¿De qué tenía que esconderse? Si se llama Juan ¿lo tendrían que llamar Francisco? cuando alguien se presenta a alguien que no ha visto nunca ¿se inventa un nombre? Realmente, y pensándolo bien, no sé como pude decir semejante tontería.

Me estuvo mostrando un apartado de ficheros que había en la BBS por supuesto, clasificados por temáticas, a las que denominaban áreas. Cada área tenía un título, y realmente había cientos de áreas, y de los más diversos temas: antivirus, documentación, utilidades de disco, programación en C, imágenes, animaciones, todas las áreas imaginables estaban allí ¡hasta áreas con ficheros para PC y Macintosh! nunca podía haber imaginado tantas áreas diferentes.

¡Pero eso no era nada! cuando se puso a dejar sus propios mensajes, vimos las áreas de mensajería. Había muchísimas, muchas más que áreas de ficheros. Le pregunté que para que servían exactamente, porque de las áreas de ficheros ya sabía algo, pero de las mensajes, no demasiado. Rápidamente me contó que eran muchos mensajes reunidos por temas, parecido a las áreas de ficheros, lo que pasa es que allí sólo había texto. Me sorprendió en un principio, y pensé, ¿quién querrá leer textos y textos porque sí? Pero mi amigo seguía contándome. Decía que lo más alucinante de los mensajes es que iban dirigidos a alguien en concreto, pero pudiendo ser leídos por todos, si así lo querías, de hecho éstos eran la mayoría. Y me mostró un mensaje dirigido a él. Tenía una cabecera que indicaba quién lo enviaba, a quién estaba dirigido y el tema sobre el que se hablaba en ese mensaje en concreto. Por ejemplo en ese mensaje dirigido a mi amigo, hablaban de los rumores de un nuevo Amiga con DSP y nuevos chips gráficos, etc. El usuario que se lo enviaba comentaba que había leído en una revista que pronto habría un nuevo Amiga, de hecho, ya había un prototipo. Mi amigo contestaba que a él le dijeron que podría llamarse A5000, y en otro mensaje más adelante, otro usuario decía que llevaría un 68060, etc. Era como una conversación, en voz alta, con un montón de usuarios a la vez. Él escribía su comentario o pregunta o lo que sea y alguien le contestaba, charlaba sobre eso con él, luego se añadían otros...

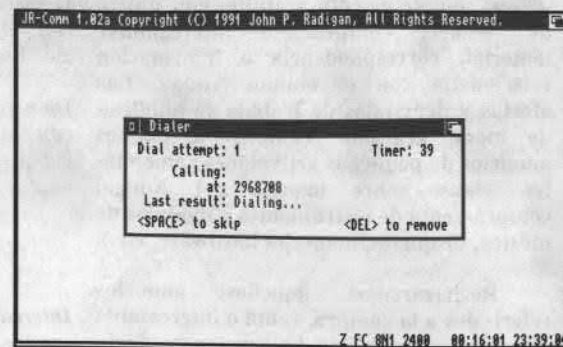
Pero, de todo, lo que me sorprendió más, fue sin duda, la amabilidad con que se trataba a todo el mundo, con una corrección y un enorme sentido del humor que en pocos sitios lo había visto. Parecía que se conocían de toda la vida y muchos ¡ni se habían visto!

Concretamente, él no conocía personalmente a nadie, pero me contó que algunos sí que se conocían de alguna ocasión: habían ido alguna vez a alguna exposición, o a prestarse alguna revista o algún libro,... Y hasta me dijo que existían unos usuarios un poco especiales, llamados, en el argot 'BBSero', puntos. Estos usuarios punto, sí que hacían reuniones regularmente, cada tres o cuatro meses, para conocerse más de cerca. Se iban a cenar o de marcha por algún sitio,... Y lo que diferenciaba a estos usuarios de los normales, los llamados de terminal, es que tenían un volumen de correo mayor que los otros usuarios, eran usuarios muy participativos, y en muchas áreas diferentes, y con más experiencia en esto de las comunicaciones. Me contó que, según le habían comentado en una respuesta a un mensaje suyo, tenían el compromiso de llamar, por lo menos, una vez a la semana, aunque, parece, que la mayoría llama cada día, y que además todos los puntos usaban un programa diferente al que usaban los otros usuarios, entre ellos mi amigo. Usaban un programa con el que se podía gestionar todo el correo muy cómodamente y ordenadamente, y que además no hacía falta teclear nada para acceder a la BBS, el programa lo hacía todo. Claro; al tener mucho volumen de correo, ¡tienen que organizarse de alguna manera!

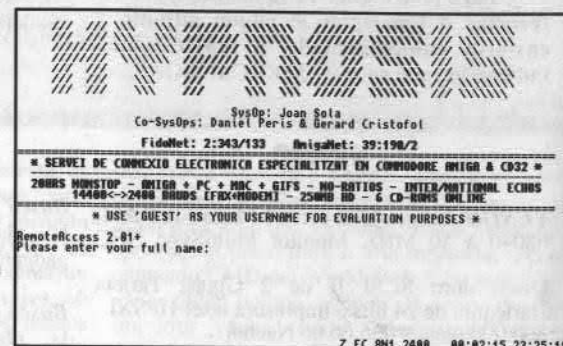
Mi amigo dejó un antivirus que consiguió de estas compañías de Dominio Público que ahora existen también aquí, y también por supuesto, dejó un par de contestaciones a mensajes que había visto escritos en el área general, donde puedes comentar cualquier cosa, rumores, saludos, anuncios, etc. Los mensajes los escribió en un editor de textos y luego los dejó en la BBS, no tuvo que escribirlos en aquel momento de la llamada.

Realmente, aquella noche salí de su casa un poco descolocado, estaba entre sorprendido y maravillado, todo lo que había visto esa tarde en casa de mi amigo me había fascinado, el descubrir que había todo un mundo escondido detrás de una corta llamada de teléfono y un pequeño aparito que producía sonidos. Todo un montón de usuarios, de compañeros e incluso de ¡amigos!, se alzaba frente a nosotros mirando el monitor atónitos, bueno yo más atónito que él. Descubrir todas las posibilidades que se abren es una experiencia difícil de describir, ¡son tantas! Tan sólo animaría a todo el mundo que lo probasen en algún momento, en casa de un amigo como yo, ¡o donde sea! Vereis que va mucho más allá de lo que normalmente se imagina todo el mundo. Pero,... si lo probais, tened cuidado,... porque esto ¡ENGANCHAS!

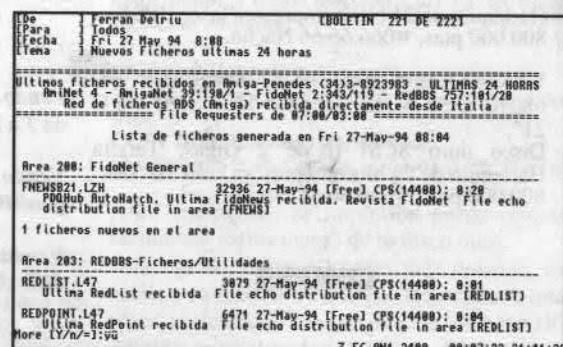
;-) ¡Saludos!



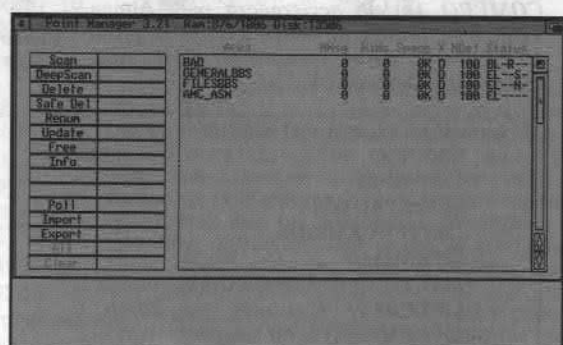
"...al seleccionar, la que era para él, su BBS habitual y preferida, el modem empezó a marcar un número de teléfono..."



"... Al entrar salió una carátula de bienvenida a la BBS, y le pidió el nombre y un apellido ..."



"... lo más alucinante de los mensajes es que iban dirigidos a alguien en concreto, pero pudiendo ser leídos por todos, si así lo querías..."



Algunos usuarios (llamados 'puntos') acceden de forma totalmente automatizada desde programas especiales, que permiten mayor acceso y velocidad al conectarse a la BBS.

En esta sección se pueden insertar anuncios personales de forma completamente gratuita, con objeto de vender, comprar o intercambiar material, correspondencia e información relacionada con el mundo Amiga. Las ofertas y demandas de trabajo se publican de modo gratuito. También aceptamos anuncios de pequeñas actividades comerciales (clases sobre manejo del Amiga, compra-venta de instrumentos o módulos de música, disquetes, montajes hardware, etc.).

Rechazaremos aquellos anuncios referidos a la compra, venta o intercambio de software pirata o cualquier otro producto cuya copia ilegal pueda violar la legislación en materia de propiedad intelectual.

Para poder insertar tu mensaje, puedes recortar o fotocopiar el cupón adjunto y enviarlo cumplimentado a AMIGA.InFo indicando en el sobre: "MERCAMIGA".

VENDO

VENDO Amiga 4000 con 52 Megas de RAM, 68040 a 50 MHz, Monitor MultiSync NEC 21". Disco duro SCSI II de 2 Gigas, Tarjeta Harlequin de 24 bits e impresora laser HPSXi. 800.000 ptas, ☎066.66.66 Nacho.

VENDO Amiga 4000 con 52 Megas de RAM, 68040 a 50 MHz, Monitor MultiSync NEC 21". Disco duro SCSI II de 2 Gigas, Tarjeta Harlequin de 24 bits e impresora laser HPSXi. 800.000 ptas, ☎066.66.66 Nacho.

VENDO Amiga 4000 con 52 Megas de RAM, 68040 a 50 MHz, Monitor MultiSync NEC 21". Disco duro SCSI II de 2 Gigas, Tarjeta Harlequin de 24 bits e impresora laser HPSXi. 800.000 ptas, ☎066.66.66 Nacho.

COMPRO

COMPRO Tarjeta aceleradora para Amiga 1200, a ser posible un 68030 a 50 MHz y con 10 Megas de RAM, copro a 60 MHz, etc... ☎066.66.66 Desesperado.

COMPRO Tarjeta aceleradora para Amiga 1200, a ser posible un 68030 a 50 MHz y con 10 Megas de RAM, copro a 60 MHz, etc... ☎066.66.66 Desesperado.

El texto de mi anuncio es el siguiente:

- ☐ VENDO
☐ COMPRO
☐ CONTACTOS
☐ INTERCAMBIOS
☐ TRABAJO
☐ ACTIVIDAD COMERCIAL
☐ CLUBS
☐ VARIOS

(Marcar en la casilla adecuada)

Contactar con _____
Dirección _____ CP _____
Población _____
Provincia _____
Teléfono y prefijo _____

TEXTO: _____

CONTACTOS

Desearía contactar con gente que haga *música con Amiga*: MIDI, Módulos, samples, etc...Escribir a Apdo. 1212 de tu pueblo.

Desearía contactar con gente que haga *música con Amiga*: MIDI, Módulos, samples, etc...Escribir a Apdo. 1212 de tu pueblo.

INTERCAMBIO

Intercambio 486 intel inside por Amiga 500, a ser posible con 512k o más. Razón: Windows me deja tirado :(☎066.66.66 PCcero.

Intercambio 486 intel inside por Amiga 500, a ser posible con 512k o más. Razón: Windows me deja tirado :(☎066.66.66 PCcero.

TRABAJO

Busco programador en C, con experiencia en la plataforma Amiga: ventanas, gadgets, bobs, etc... ☎066.66.66 Nacho.

Busco programador en C, con experiencia en la plataforma Amiga: ventanas, gadgets, bobs, etc... ☎066.66.66 Nacho.

ACTIVIDAD COMERCIAL

Vendo disquetes vírgenes de 5 a 200 ptas, de 3 a 20 ptas ☎066.66.66 Alberto.

Vendo discos de dominio público a 400 ptas. ☎066.66.66 Atanasio.

Vendo disquetes vírgenes de 5 a 200 ptas, de 3 a 20 ptas ☎066.66.66 Alberto.

Vendo discos de dominio público a 400 ptas. ☎066.66.66 Atanasio.

Vendo disquetes vírgenes de 5 a 200 ptas, de 3 a 20 ptas ☎066.66.66 Alberto.

Vendo discos de dominio público a 400 ptas. ☎066.66.66 Atanasio.

VARIOS

Vendo lentillas usadas, por no usar. Sólo tienen 4 años, han pasado de blandas a duras y han cambiado de color (ahora son negras de tanto mirar Windows). ☎066.66.66 PCcero.

Vendo lentillas usadas, por no usar. Sólo tienen 4 años, han pasado de blandas a duras y han cambiado de color (ahora son negras de tanto mirar Windows). ☎066.66.66 PCcero.

Vendo lentillas usadas, por no usar. Sólo tienen 4 años, han pasado de blandas a duras y han cambiado de color (ahora son negras de tanto mirar Windows). ☎066.66.66 PCcero.

CLUBS

Club Anastasio.Miguel.Isidro.German.Antonio somos un grupo de amigueros interesados en formar un club en Vallapalos Quemedieron, con el nombre anterior. Escribir al Apdo. 1212 de nuestro pueblo.

Club Anastasio.Miguel.Isidro.German.Antonio somos un grupo de amigueros interesados en formar un club en Vallapalos Quemedieron, con el nombre anterior. Escribir al Apdo. 1212 de nuestro pueblo.

Club Anastasio.Miguel.Isidro.German.Antonio somos un grupo de amigueros interesados en formar un club en Vallapalos Quemedieron, con el nombre anterior. Escribir al Apdo. 1212 de nuestro pueblo.

Club Anastasio.Miguel.Isidro.German.Antonio somos un grupo de amigueros interesados en formar un club en Vallapalos Quemedieron, con el nombre anterior. Escribir al Apdo. 1212 de nuestro pueblo.

Club Anastasio.Miguel.Isidro.German.Antonio somos un grupo de amigueros interesados en formar un club en Vallapalos Quemedieron, con el nombre anterior. Escribir al Apdo. 1212 de nuestro pueblo.

Club Anastasio.Miguel.Isidro.German.Antonio somos un grupo de amigueros interesados en formar un club en Vallapalos Quemedieron, con el nombre anterior. Escribir al Apdo. 1212 de nuestro pueblo.

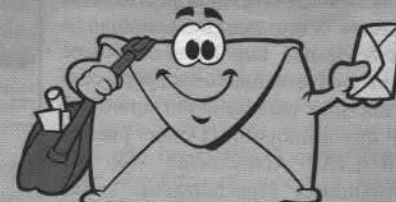
Enviar este cupón a:

AMIGA.InFo (REF.MERCAMIGA)
Apdo. de Correos 166
08750 Molins de Rei
Barcelona

Bienvenidos a la sección de Cartas y Dudas, donde cada mes nuestro equipo de técnicos de Amiga.InFo intentará contestar y resolver vuestros problemas y dudas, y donde podréis reflejar vuestras opiniones, críticas, sugerencias, etc., relacionadas con cualquier tema de Amiga o la revista. Nuestra intención es publicar todas y cada una de vuestras cartas, puesto que sabemos que la mayoría de problemas son comunes a todos los usuarios. No te asustes si tus preguntas pueden parecer sencillas o muy complicadas, si no entiendes o sabes mucho, ¡escribenos! Para ponerse en contacto con esta sección envía tu carta a:



AMIGA.InFo (REF:Cartas)
APDO. 166
08750 Molins de Rei
BARCELONA



ATENCION: Durante este primer mes, y como es lógico, no hemos tenido correspondencia, pero hemos estado recogiendo todo tipo de problemas, opiniones, sugerencias, etc., entre los redactores y colaboradores para montar este primer número de AInFo. Por supuesto, la redacción no se hace responsable del contenido de ninguna carta.



CHIP, FAST Y HAM...

- 1) ¿Qué diferencia hay entre la memoria FAST y la CHIP? ¿Para qué sirven?
- 2) ¿Cómo funciona el modo gráfico HAM del Amiga? ¿Es posible usar 16 millones de colores simultáneamente en pantalla, en modo AGA?

Bueno, vayamos por partes:

1- La memoria FAST (rápida) es usada exclusivamente por el procesador central (Motorola 680x0) para acceder a datos e instrucciones de programa. La memoria CHIP es utilizada por los coprocesadores gráficos, de sonido y el propio procesador central para almacenar datos de gráficos, sonido y sistema operativo. Debido a la cantidad de accesos simultáneos que debe soportar esta memoria de todos los coprocesadores, el tiempo de respuesta es más lento. Generalmente tienen prioridad los coprocesadores frente al procesador, y es por esto que los programas se vuelven más lentos cuando sólo hay instalada memoria CHIP. Es muy conveniente tener como mínimo dos megas de FAST para obtener la máxima velocidad de ejecución de los programas. Si, por ejemplo, tienes un A1200, el cual posee dos megas de CHIP instalados de fábrica, has de saber que con la instalación de memoria FAST tu ordenador es aproximadamente el doble de rápido que otro A1200 sin FAST.

2- El modo gráfico HAM es una forma "económica" de representar muchos colores con pocos bits. Las siglas significan HOLD AND MODIFY ("agarrar y modificar"). La potencia de este modo se basa en una especie de direccionamiento relativo en la forma de codificar los colores. Cada pixel tiene el color codificado en bits absolutos dentro de los registros de colores, o sólo los incrementos respecto al pixel anterior. Esta segunda forma es la que permite economizar memoria, ya que sólo hay que guardar la diferencia de colores entre pixels contiguos. Respecto a representar 16 millones de colores

simultáneamente en pantalla la respuesta es NO. No por limitación de hardware, sino porque en una pantalla de por ejemplo 1000x1000 pixels, sólo hay 1.000.000 de pixels y cada uno de ellos sólo puede tener un color, por lo que nunca se utilizan los 16 millones a la vez. Sin embargo, los modos HAM6 y HAM8 comentados antes sólo utilizan 4096 y 262.144 colores, y se pueden representar simultáneamente en pantalla, puesto que hay como mínimo tantos pixels como colores diferentes (muy listos estos ingenieros de Amiga...). Otro tema es cambiar la paleta de colores dinámicamente con el procesador central, como el método utilizado por NewTek llamado "Modo Dynamic Hires".



HP DESKJET 520...

Me he comprado recientemente una impresora Hewlett Packard Deskjet 520, y quiero utilizarla con mi Amiga 2000, pero no encuentro el driver propio de ella, y de momento sólo puedo imprimir utilizando el driver de la 510. ¿Sabéis dónde puedo conseguirlo? Para imprimir utilizo procesadores de texto, y un programa de autoedición, pero este último es increíblemente lento (hasta 10 minutos por página!). ¿Sabéis cómo puedo mejorar la velocidad de impresión?

TODOS los drivers de las impresoras de las marcas más importantes de fabricantes como HP, Canon, Epson, Brother, Citizen, Facit, Fargo, Fujitsu, Nec, Oki, Panasonic, MannesMann Tally, Sekoisha, Siemens y Star, con sus correspondientes versiones, están disponibles a través de dos programas diferentes: TURBOPRINT, de IrseeSoft, o STUDIO PRINTER. Está concretamente el que necesitas, HPDeskjet520. Estos programas mejoran en gran medida tanto la velocidad como la calidad de impresión, obteniendo el máximo de cada impresora. Para conseguirlos ponte en contacto con tu distribuidor de Amiga habitual y pregunta por ellos. Aparte de utilizar estos programas para

mejorar la velocidad, puedes emplear el truco de no enviar directamente el fichero a la impresora, sino que lo redireccionas hacia, por ejemplo, el disco duro o a la memoria, con el comando CMD del Workbench. Con este truco ganas algún tiempo, además de que si quieres imprimir varias copias del mismo fichero, únicamente tienes que copiar el fichero creado por CMD a la impresora tantas veces como quieras, por ejemplo: Antes de imprimir con tu programa de autoedición, y desde una SHELL/CLI, teclea el siguiente comando:

CMD Parallel DH0:Tufichero.impre (y cuando acabe de imprimir escribes en la shell...)
COPY DH0:Tufichero.impre PAR: (y si quieres más copias, vuelves a escribirlo)

¡Ojo! Un fichero de impresión puede ocupar fácilmente varios megas de tu disco duro. Otra opción, aunque mucho más drástica, es acelerar tu A2000, ya que por ejemplo con una placa aceleradora con un procesador M68030 /40Mhz puede tardarte menos de la mitad como mínimo.



ESTOOO... ¿ES LA CIA? ¿MANDE?...

Tengo un problema con el puerto paralelo de mi A500, ya que desde hace dos días me es imposible utilizar ni la impresora, ni el digitalizador de audio, ni nada que se tenga que conectar en el susodicho puerto. Se lo he comentado a mis amigos, pero cada uno me da soluciones diferentes... uno dice que es del CIA (¿qué es eso? No tiene nada que ver con el centro de inteligencia... ¿verdad?), otro que son los periféricos, otro que mi A500 está para cambiarse por un A1200, y yo, que no sé qué hacer.

Es muy frecuente en los usuarios que utilizan a menudo el puerto Paralelo o Centronics, con diferentes periféricos, tengan problemas de este tipo. La causa: el CIA, o Complex Interface Adapter. Posiblemente, uno de tus periféricos ha sobrecargado o afectado a este

chip, que, entre otras cosas, se dedica a controlar el puerto paralelo. La solución es fácil: sacar el chip y cambiarlo. En un A500 se identifica porque lleva el número 8520 y en la placa se identifica por el componente U7 o ODD CIA (A). No te asustes, cambiarlo es simplemente sacarlo del zócalo utilizando un pequeño destornillador y colocar un CIA nuevo. Por cierto, si no utilizas el puerto serie para nada, a veces ocurre que, intercambiando los dos CIA 8520 que hay en el A500, el puerto paralelo vuelve a funcionar...pero esto es una solución sólo para salir del apuro, al final tendrás que cambiarlo ¿Dónde puedes conseguir un CIA? Actualmente sólo en algunos distribuidores especializados, como puedan ser CIB en Madrid, CRE en Vizcaya o Amiga Center en Barcelona.

✓ PC 486 DESGUAZADO...

Tengo una unidad de alta densidad de un PC 486 que estoy desguazando después de haberme comprado un A4000, y también un disco duro IDE de 260 megas, así como un monitor multisync y 4 mb de RAM tipo SIMM 72 pins. ¿Puedo aprovechar algo para el A4000?

¡Vaya cambio afortunado! Un 486 por un A4000... Tanto la RAM, como el disco duro, como el monitor multisync (sobre todo si trabaja con las frecuencias de 15khz en adelante) pueden ser utilizados sin ningún tipo de problemas y, además, se instalan directamente en el A4000. En cuanto a la unidad de alta densidad, es muy extraño que te funcione puesto que las que utiliza Commodore son de la marca Chinon y difíciles de encontrar en un PC. Estas unidades están preparadas para funcionar con el sistema operativo del Amiga y las de un PC no.

✓ CHECKSUM ERROR ON BLOCK... ARRG!!! BLOCK OUT OF RANGE... PIF!

Hola, me dedico a la Infografía con el Amiga y por precaución siempre guardo mis trabajos en "backups" del disco duro. Pero un día, mientras estaba salvando una animación 3D en el disco duro, el fluido eléctrico se cortó, y ahora, cada vez que enciendo el Amiga, al cabo de unos minutos de intensa actividad del disco duro me aparece un mensaje del sistema que me dice algo así: "Error validating unit Dh1. Block 2158155 out of range" y la opción de reintentar o cancelar. Si le doy al reintentar me vuelve a salir el mensaje, y si le doy al cancelar, no, pero no puedo grabar nada en la partición Dh1 del disco duro. Los datos están, puesto que los puedo leer y copiar, pero no puedo grabar nuevos, además, la partición aparece como llena al completo, cuando estaba tan solo un 40 % antes del corte. ¿Cómo puedo solucionarlo? ¿He perdido algún dato? ¿Se ha estropeado el Amiga?

No sólo ocurre cuando hay apagones, también cuando se 'cuelga' el Amiga, o interrumpes de una forma brusca (reset) en el momento en que estás salvando datos en el disco duro (o unidad de disco). Tu problema se llama "error de



UNA CRITICA DE UN USUARIO... QUE NO TIENE PELOS EN LA LENGUA

Mediados de 1992. Commodore se ha dormido en los laureles y sigue viviendo del éxito del Amiga, sin hacer otra cosa que pequeños cambios, la mayoría estéticos, que lo único que consiguen es crear confusión entre la mayoría de los usuarios y potenciales compradores (500+ y 600, por un lado, y el 3000, su único acierto, aunque muy caro). Mientras tanto, el Amiga, que cuando salió al mercado era indiscutiblemente el mejor ordenador personal del momento, ve como sus rivales naturales, el PC, y las consolas, no sólo se han puesto a la altura de éste, sino que incluso lo han superado en algunos aspectos, y entonces ocurre lo previsible: el Amiga ya no marca la diferencia, y empieza el fluir de distribuidores, usuarios, casas de software, etc. hacia los nuevos mercados, al no ver futuro en el Amiga.

Mientras tanto, Commodore no parece dispuesta a reaccionar, y no reconoce que ha perdido la oportunidad de controlar casi totalmente el mercado del ordenador personal, al no ser capaz de introducir en el mercado un nuevo Amiga, mientras que tampoco ha conseguido que el Amiga sea reconocido como algo más que una consola con teclado en Europa (aunque por esa fama se hayan vendido la mayoría de los Amigas), mientras que en USA ocurre lo contrario (sobre todo gracias al Video Toaster), ganando gran parte del mercado de vídeo, pero despreciando inexplicablemente el potencial del Amiga como consola, y perdiendo casi completamente el mercado USA (quedando el Amiga casi como un ordenador para vídeo).

Viendo esta situación, parece que los directivos de Commodore se han hundido a sí mismos y prefieren vivir de rentas mientras éstas duren, pero a finales de año ocurre por fin lo esperado, y salen al mercado dos nuevos Amiga, el 4000 y el 1200, que devuelven el protagonismo tecnológico al Amiga (arquitectura 32 bits, modos gráficos AGA, CPU de 32 bits).

1993 se presenta con los dos nuevos Amiga, pero aunque Commodore ahora ya tiene con que contrarrestar a la competencia, la crisis le golpea con fuerza y esto le fuerza a dejar gran parte de sus mercados (ganados gracias a la calidad del Amiga y de sus usuarios, los mejores publicistas que podía encontrar Commodore), a merced de sus competidores, como es el caso de España.

A finales de 1993, Commodore lanza su segunda tentativa CD, el CD-32 (la primera, CDTV, fracaso gracias de nuevo a las indecisiones de Commodore, incapaz de decidir si lanzarlo como parte de la gama Amiga o como un nuevo tipo de vídeo), con lo que se adelanta a sus competidores en la carrera por lanzar consolas CD de 32 bits (una carrera que todavía continúa). Este lanzamiento consigue devolver el protagonismo de nuevo al Amiga, ya que la mayor parte de las casas de software intentan subirse al carro del CD (debido sobre todo al ser un formato no pirateable, al menos en teoría...), consiguiendo arrastrar también al 1200 a los puestos punteros en las ventas.

Mientras todo parece indicar un resurgimiento del Amiga, ¿que sucede en España?. Aquí, la fiebre de los PC arrastra a todo el mundo, y el Amiga, casi sin distribuidores, sin manuales, sin apoyo de Commodore y, sobre todo, sin "juegos" espectaculares no puede competir y sólo los fieles seguidores compran los nuevos Amiga.

Pero en el horizonte se atisba un rayo de luz, ya que esta situación cambiará próximamente (ya está cambiando, con nuevas tiendas que apoyan al Amiga surgiendo en los lugares más insospechados) debido a que Commodore va a volver a tener distribuidores oficiales y servicio técnico en España, una decisión que puede volver a dar al Amiga un protagonismo que nunca debió perder en este país. Este apoyo debe servir para que por fin el Amiga tenga el reconocimiento que se merece, aunque todo depende, en gran parte, de los distribuidores, que deben mirar hacia el futuro y no matar al Amiga (léase precios abusivos, nula asistencia técnica, manuales en alemán...), ya que los usuarios bastante han hecho ya (siendo los más fieles, e intentando siempre dar a conocer el Amiga, siendo esta misma revista un ejemplo).

Para terminar, si no te han convencido estos argumentos, tú te lo pierdes, porque mientras haya software para Amiga como Scala, Directory Opus, LightWave, y PageStream siempre habrá fieles seguidores del Amiga.

Un fiel seguidor.

NOTA DE REDACCION: Esta carta es una carta real escrita por un usuario de Amiga cuando se enteró de que se publicaría AInFo. Nos la dio mucho antes de producirse las recientes noticias de que Commodore US había entrado en proceso de liquidación. En cuanto a que Commodore ha vuelto a España, nosotros no tenemos ninguna noticia y es bastante improbable que vuelva, al menos bajo ese nombre (Leer la primera página de noticias). No podemos afirmar que exista un servicio técnico proporcionado por la empresa Eltec y menos que cubra todo el territorio nacional.

De nuevo, queremos avisar de que la redacción publicará cualquier carta recibida sin modificar su contenido, y que AInFo no se hace responsable de las opiniones vertidas en esta sección.

validación". Generalmente se debe a que la información queda corrupta (no se ha finalizado la escritura) en el disco, y el mapa de ficheros del disco ha sido modificado. En estos casos, cuando vuelves a conectar tu Amiga, el sistema operativo chequea (valida) el disco duro y comprueba que está corrupto, e intenta solucionarlo reordenando y haciendo desaparecer el fichero que ha ocasionado el desperfecto. Pero lo arregla un 80% de las veces, y a ti te ha tocado el 20% donde están los errores graves...

Te diría que la solución más fácil es hacer un backup de la partición, formatearla de nuevo, y restaurar el backup. Así desaparece el problema. Sin embargo, aparece otro: si tienes una partición de, por ejemplo, 300 Megs, ¿cuántos discos necesitas para hacer un backup?, ¿cuánto tiempo?

Bueno, necesitarías unos 350 discos y algo así como 12 horas sin parar de cambiar discos... Si es éste tu caso no te asustes, ya que esta vez va a ser una utilidad de dominio público la que te puede salvar. Se llama "DiskSalv" y está programada por Dave Hayne (un genio del Amiga). Esta utilidad la deben tener todos los usuarios de disco duro precavidos, ya que recupera ficheros borrados, desformatea, salva, valida y repara discos duros con problemas de tu tipo. Es muy eficiente en la mayoría de los casos. Puedes conseguirla en cualquier casa de dominio público, o BBS, o en el disco del mes de este número de AInFo.

✓ CD32 Y A1200 ¿SE ENTIENDEN?

Quiero conectar una CD32 a un A1200 a través del puerto serie o del puerto paralelo. ¿Dónde puedo encontrar los cables y el programa? Si conecto la CD32, ¿podré utilizar el CDROM que lleva incorporado en mi A1200? Es decir, ¿podré acceder, vía serie o paralelo, a los ficheros almacenados en CDs desde el A1200?

La CD32 no dispone de puertos paralelo o serie, pero existen algunas empresas que han diseñado interfaces para la CD32 que permiten conectar unidades de disco, teclado, ratón y tienen puerto serie y paralelo. Pregunta a tu distribuidor Amiga habitual para más información.

En cuanto tengas alguno de los dos puertos, los programas de dominio público SERNET y PARNET te permitirán acceder sin ningún problema al CDROM de la CD32, y los cables son los típicos de módem nulo u ordenador-ordenador por el paralelo, que puedes conseguir en cualquier tienda de informática o incluso hacértelos tú mismo. (Junto con los programas SERNET y PARNET vienen los esquemas del conexionado del cable).

✓ A3000 para 3D y Autoedición

Tengo un Amiga 3000 con 4 megas de ram y 340 Mb de disco duro. Hasta ahora no he trabajado en serio con él, pero voy a empezar a introducirme en el mundo de los gráficos 3D y de la autoedición. Quisiera hacerles algunas

preguntas para averiguar que hardware y software es el recomendado por Vds. para estos temas:

- a - ¿Qué programas de 3D están disponibles en el mercado para Amiga?
- b - ¿Qué memoria es necesaria para trabajar adecuadamente?
- c - ¿Serían muy lentos los "renders" en mi A3000 sin acelerar?
- d - ¿Qué programas de autoedición me recomiendan?
- e - ¿Qué impresoras de chorro de tinta o láser trabajan con Amiga?
- f - ¿Sería necesario un scanner, o con un digitalizador de 24 bits ya tendría suficiente?
- g - ¿Sería necesario cambiar el monitor para que no "parpadee" en alta resolución? ¿Sería conveniente una tarjeta 24 bits?

a) Actualmente hay muchísimos programas de Amiga dedicados a los gráficos 3D (quizás es el aspecto más avanzado que Amiga tiene frente a cualquier otra plataforma) y la lista de ellos es bastante extensa. Pero supongo que te interesan los que más éxito han tenido y los más potentes: LightWave, Imagine y Real 3D. Estos tres programas reúnen lo mejor en creación y animación de gráficos generados por ordenador (con ellos se han hecho los efectos especiales de películas muy conocidas) y cada uno tiene sus ventajas y desventajas (si es que se pueden llamar así). El más reciente ha tenido una aceptación muy importante, se trata de LightWave 3.5 (lee el artículo de la Video Toaster de este mes para más información), aunque también está dando muy fuerte la versión 3.0 de Imagine. Todos ellos están disponibles a través de cualquier distribuidor de Amiga (Si tienes problemas para encontrarlos, PiXelSoft, tiene a la venta el software LightWave 3.5 y GaliFrame el Imagine 3.0)

b) Para trabajar bien necesitas como mínimo 10 Mb de RAM tanto en autoedición como en 3D; para trabajar lo que se dice en serio, de 16 megas en adelante. Como anécdota, comentarte que hemos visto un caso de un A4000 y 32 mb de RAM que está cada dos por tres "out of memory" debido a que su dueño lo utiliza para crear animaciones 3D profesionales...y muy impresionantes.

c) La velocidad para rendear con un A3000 es aceptable, y más que aceptable si lo comparas con los otros modelos de Amiga (excepto el A4000, claro), aunque siempre tienes la opción de acelerarlo con alguna tarjeta como la de GVP 040, y colocar tu Amiga por delante de cualquier otro. Para autoedición es suficiente.

d) Los programas de autoedición que actualmente destacan son el PageStream y Professional Page, y procesadores de texto potentes son el Final Writer, Final Copy II o WordWorth. Concretamente el Professional Page ha finalizado su desarrollo en la versión 4.1 (esta revista está montada con la versión 4.0) y la empresa Gold Disk ha dejado por el momento de trabajar en él. Por otro lado, se espera desde hace tiempo la llegada inminente del que, según sus programadores en SoftLogik, va a ser la opción en autoedición

más potente del mercado, por encima de Apple Mac y PC, la versión 3.0 de PageStream. La verdad es que, si realmente incorpora todas las características profesionales que prometen, va a ser un duro golpe para todos sus competidores, puesto que está implementado en Amiga y aprovechando todas sus prestaciones al máximo. En Amiga.InFo estamos a la espera de su llegada a España, y en cuanto tengamos oportunidad lo analizaremos a fondo. PiXelSoft ha anunciado una versión en castellano.

e) En teoría, cualquier impresora puede funcionar con Amiga, y digo en teoría porque para ello hace falta el driver correspondiente que controle el modelo determinado de impresora. Afortunadamente, existen decenas de drivers de las impresoras más utilizadas (y además suelen ser de dominio público), por lo que no hay problema en este aspecto. En cuanto a modelos concretos, nosotros en AInFo hemos utilizado una CANON BJ10ex, una HP520 y una HP Laserjet 4P. Las dos primeras dan excelentes resultados y son de inyección de tinta, mientras que la tercera es una laser con resolución de 600dpi/600dpi que da un resultado profesional. Todas han sido utilizadas con Amiga sin ningún problema.

f) Aparte de la diferencia de precio (un digitalizador es mucho más económico que un escáner) está la diferencia de calidad. Para resultados realmente profesionales necesitas un escáner color 24 bits, con resoluciones de 600x600 en adelante (el EPSON GT6500 Color, por ejemplo) y un programa como Art Department Professional que permita controlarlo. Sin embargo, si lo que te interesa es la economía, y quieres resultados normales, un digitalizador de 24 bits en tiempo real (por ejemplo, el VLAB) ya es suficiente. La diferencia entre una imagen escaneada y una digitalizada es abismal, pero a veces el precio también.

g) Es muy conveniente que adquieras un monitor multisync o un SVGA, ya que las mayores resoluciones tanto en 3D como en autoedición no parpadean en éstos. Afortunadamente el A3000 está preparado para trabajar con este tipo de monitores directamente.

Una tarjeta de 24bits (Picasso, Retina, etc...) y monitor multisync (Commodore 1942 o 1960) son la pareja ideal para obtener resultados profesionales en 3D y trabajar cómodamente en autoedición.

✓ DISCO DURO 3.5", PCMCIA y CDROM

Tengo un A1200 y quisiera saber si es posible instalarle un disco duro de 3.5" IDE en su interior, y si es así, qué marca me recomiendan y qué es necesario para ello. También poseo una tarjeta de memoria PCMCIA de 2mb y me han dicho que este tipo de tarjetas baja el rendimiento del ordenador. ¿Es eso cierto? Por último, quisiera preguntarles si existen CDROMs para el A1200 y cuáles. Gracias por todo.

Sí es posible instalar un disco duro de 3.5" en un A1200, y para ello es necesario ser un poco manitas. Entre nosotros hay quienes lo tienen en el interior del A1200 y quienes lo han

instalado en una caja exterior. Se necesitan un par de cables, uno que pase del conector de 44 pins de un HD 2.5" a los 40 pins de un 3.5", y otro para extraer la alimentación del conector de la unidad de disco. Los cables están disponibles, por ejemplo, en Amiga Center y CIB. Las marcas de HD 3.5" más utilizadas en Amiga son Seagate y Conner, puesto que no dan ningún problema (Quantum puede dar problemas en algunos modelos).

Es cierto que las tarjetas PCMCIA de memoria disminuyen el rendimiento del A1200. Esto es debido a que estas tarjetas tienen memorias de 16 bits, mientras que el A1200 es una máquina de 32 bits, y para leer/escribir un dato en esta memoria se necesita realizar dos accesos (16 bits en el primero y los otros 16 bits en el siguiente acceso) frente a 1 acceso cuando accede a memoria de 32 bits. Consecuencia: el procesador ha de esperar a la memoria dos veces por acceso y los programas van más lentos (En test de AIBB un A1200 con PCMCIA es un ¡¡40% más lento que el original sin expandir!!). Ocurre todo lo contrario si utilizas memoria FAST de 32 bits en el puerto interno del A1200, puesto que se acelera hasta el doble un A1200. Moraleja: no compres una memoria PCMCIA, compra RAM 32 bits.

Todavía no se distribuyen (y no tenemos noticia de que se haga en un futuro inmediato), pero Commodore ha sacado un CDROM para el A1200. Es prácticamente el mismo que el incluido en la CD32 y además incorpora el chip AKIKO, que permite al A1200 compatibilizar al 100% los juegos de esta magnífica consola. También es posible instalar internamente en el mismo CDROM 4 megas de tipo SIMM. Por otro lado, en Inglaterra también se comercializan CDROMs conectados por el PCMCIA del A1200 que ofrecen prestaciones similares al de Commodore.

8 CANALES Y DOS AMIGA

Soy aficionado al Amiga por su excelente sonido y los programas de música que existen para él. Recientemente me han comentado que existe un "Protracker" que permite conectar dos Amigas y obtener ocho canales simultáneos... ¿Es cierto? ¿Cómo se conectan? ¿Sirve cualquier modelo de Amiga? Tengo un viejo A500 y un A4000 y me gustaría poder aprovecharlos al máximo en sonido.

Es cierto que existe un "Protracker" que permite conectar dos Amigas para utilizar 8 canales simultáneos. En nuestra redacción tenemos concretamente la versión "Protracker 1.2e (DUAL VERSION)", de dominio público, y que apuntamos ahora que en el próximo número de InFo ha de incluirse en el disco del mes. Cuando se ejecutan en dos Amigas conectados con un cable de modem nulo, los "Protrackers" se envían comandos por el puerto serie a una velocidad de 19200 baudios. Desde uno de ellos puedes controlar al otro, e incluso componer módulos de 8 canales utilizando únicamente un teclado. Sirve cualquier modelo de Amiga, por lo tanto podrás utilizar tu A500 y el A4000 perfectamente.



¿INTERNET EN ESPAÑA?

Últimamente se habla mucho de Internet, y dentro de Internet de Aminet, donde al parecer existen gigas de programas y datos de Amiga. He estado buscando información de cómo conectarse, pero no he encontrado nada. Tengo un modem de 19200 baudios y suelo conectarme a muchas BBS's. ¿Sabéis si puedo utilizarlo para conectarme a Internet? ¿Hay en España algún servicio como Internet? ¿A quién puedo dirigirme para conectarme a esta red internacional?

En España, el tema de Internet está muy verde todavía. En otros países existen empresas privadas a las que se les solicita que te instalen el servicio de conexión a Internet (como si te instalaran un teléfono) y te dan de alta para acceder a toda la información que existe (no sólo de Amiga). Aquí el único método asequible que existe para conectarse, ya que contratar este servicio a telefónica para uso particular es carísimo, es pertenecer a una Universidad (es decir, ser estudiante) y tener acceso a un terminal que te permita conectarte con otras Universidades y con Internet. Es una verdadera lástima, porque por esta red internacional pasa diariamente una cantidad increíble de información de todo tipo, y por supuesto de Amiga. En InFo tenemos acceso a ella a través de algunos de nuestros redactores y os mantendremos constantemente informados de las noticias y programas interesantes que vayan surgiendo.

Por otro lado, y también en el extranjero, hay BBS's que actúan como puertas a Internet, puesto que al conectarte a ellas (con una llamada normal de teléfono) accedes a las áreas de Internet de las cuales son "mirror", es decir, tienen una copia actualizada diariamente y de forma automática de todos los ficheros que por ejemplo existen en Aminet. Aquí en España este servicio de BBS es nulo y es debido únicamente a los precios de Telefónica. Como máximo, algunas BBS nacionales importan las noticias y mensajes electrónicos de Internet, y algunas también permiten enviar algún mensaje al exterior, pero, generalmente, utilizan la red FIDONET para ello. Aún estamos en la edad de piedra en este tema, otros países nos llevan como mínimo diez años de ventaja.



¿MONITORES SVGA CON AGA?

¿Puedo conectar un monitor SVGA para ver los modos gráficos AGA nuevos sin flicker? ¿Son mejores estos monitores que el Commodore 1942?

Es posible conectar a tu A4000, A3000 o A1200 un monitor SVGA con tan solo utilizar el cable y adaptador correspondiente a salida SVGA (conector tipo joystick). Sin embargo, es poco recomendable, puesto que estos monitores suelen tener la velocidad de sincronismo horizontal a partir de los 31.2 KHz y el Amiga trabaja en su modo nativo a 15.6 KHz. ¿Qué significa esto? Pues que cuando, por ejemplo, ejecutes programas que trabajen en modos PAL normales de Amiga no aparecerá nada de nada en tu monitor SVGA (y hay cientos de ellos, como, por ejemplo,

programas de videoedición, todos los juegos, demós, etc.). Si únicamente quieres utilizar los modos gráficos sin flicker (doublePAL a 72 Hz...) no tendrás problemas.

¿Es una desventaja que el Amiga trabaje a 15.6 khz mientras los demás ordenadores no lo hacen?

No. Todo lo contrario, puesto que todos los genlocks, digitalizadores y la mayoría de aparatos del mundo del vídeo trabajan con esa frecuencia. Es carísimo para un PC poder adaptar un genlock, o simplemente grabar la salida de su ordenador directamente en vídeo. No pueden. Trabajan a 31.2 KHz. Esta es una de las razones de que el Amiga sea por excelencia el ordenador más aplicado en el mundo del vídeo.

Los monitores SVGA suelen ser de las mismas características que los de Commodore, pero con la diferencia de que éstos últimos trabajan con todos los modos de pantalla del Amiga. Es una elección mucho más sabia comprarse un C1942 o un monitor multisync, siempre que tenga el mínimo en 15.6 KHz.



GRACIAS, GRACIAS...

Enhorabuena por esta revista. Iniciativas como ésta son las que hacen a los usuarios de Amiga diferentes en la forma de pensar cuando hablan de un ordenador. Nosotros hablamos de música, sonido, imagen, componer, formar, iniciar, programar..., hablamos en definitiva de "crear", sea cual sea el campo en el que apliquemos el Amiga. Los demás usuarios de otro tipo de ordenadores hablan de bases de datos, cuentas, megaherzios y programas fríos y calculadores... Gritemos "¡NO!". El hombre es un ser creativo y los ordenadores tienen que servir a esta idea. De momento, el Amiga es el único que va en esa dirección, y si desaparece, creo sinceramente que no hay alternativa que lo iguale, y lo peor de todo es que no la hay tampoco en el futuro...

Únicamente he visto la versión aún no finalizada de Amiga.InFo, pero me parece genial y espero que continuéis en esta línea sería pero distendida, y sobre todo que habléis de todos los temas de Amiga. La idea de dar más importancia a los distribuidores nacionales y prestarles atención es muy interesante, espero que ellos colaboren tanto como lo hacéis vosotros, puesto que el resultado es un beneficio mutuo entre el usuario final y el propio distribuidor, a través de vuestra revista. Finalmente quisiera saber como colaborar con la revista, a dónde debo dirigirme y si puedo escribir algún artículo... Gracias y ¡suerte!

Ante todo decirte que compartimos tu forma de pensar y la idea que tienes de nuestro Amiga. En esa idea se basa toda la revista.

Por supuesto que puedes colaborar con nosotros, ya sea escribiendo artículos, mandando cartas, enviándonos tus ideas, programas, trabajos o lo que quieras.:

Amiga. InFo ADPO. 166
08750 MOLINS DE REI (BARCELONA)

Este artículo quiere introducir brevemente un cursillo sobre el lenguaje de programación AMOS Basic y explicar las posibilidades que ofrece. Antes de todo, una pequeña aclaración: actualmente hay disponibles el AMOS y el AMOSPRO, pero en el texto me refiero a ambos con la palabra AMOS, excepto si hago mención expresa de lo contrario.

El AMOS fue creado en 1990 por François Lionet, un programador francés que había creado para los ATARI-ST un nuevo lenguaje llamado STOS. El STOS se caracterizaba por un aprovechamiento excelente de los gráficos y sonido de los ATARI-ST, y estaba pensado para crear juegos y demostraciones. Viendo de donde procede, es lógico pensar que estaba principalmente pensado para programadores de juegos y demostraciones. Posteriormente ha ido mucho más allá, y con el AMOSPro versión 1.0, aparecido en 1992, se ha convertido en un lenguaje que permite crear casi cualquier programa que se nos ocurra.

Un detalle que hay que agradecer a François Lionet, es que no hace falta tener una copia original de AMOS para trabajar con él. Cuando alguien lo compra, está pagando los manuales de instrucciones y ejemplos, y no el permiso de uso. Si un usuario quiere comercializar un producto desarrollado en AMOS, entonces tiene que comunicarlo y pagar la licencia a EuroPress Software, los cuales son los proveedores y distribuidores del lenguaje en Europa. Pero que nadie se asuste, pues el precio de la licencia es muy bajo. De todas maneras si está interesado en hacer algún programa, le recomiendo que compre la versión original de AMOSPro versión 2.0 (la última por ahora) la cual consta de 7 discos, menú de ayuda en editor, manuales, ejemplos y un programa que permitirá convertir nuestros programas en ejecutables (el compiler), sin necesidad de cargar siempre el AMOSPro para hacerlos funcionar. Todo ello aproximadamente por unas 20.000 pesetas.

Si quiere tenerlo gratis y no le interesan para nada los manuales, puede conseguir las versiones anteriores en casi cualquier BBS de AMIGA (está actualmente disponible la versión 1.36 aunque sin el compiler). Un último comentario: el compiler no es un programa de dominio público y tiene que comprar el programa original. De todas maneras, le recomiendo que se haga con el programa original AMOSPro y el compiler correspondiente.

Es un lenguaje de programación tipo basic, y, curiosamente, la mayoría de los comandos típicos del basic los ha heredado del Sinclair Basic del ZX Spectrum. Lógicamente ha incorporado muchos comandos nuevos que aprovechan las capacidades del Commodore Amiga, sobre todo en lo que se refiere a gráficos. Esta herencia de comandos permitirá comprender más fácilmente el uso de algunos de ellos a los antiguos poseedores de Spectrum.

Este lenguaje se caracteriza por poseer un juego de instrucciones potente y fácil de aprender y es razonablemente rápido para ser un basic. Con AMOS es perfectamente posible

hacer aplicaciones, demostraciones de música, animaciones y juegos. Por contra, no aprovecha para nada el sistema operativo y no podemos crear aplicaciones con ventanas y menús de estilo "Intuition", aunque permite crear todo esto, pero con su estilo propio. Esta falta de entendimiento con el sistema operativo conlleva otro problema muy importante: no saca partido de las resoluciones ECS y mucho menos las resoluciones AGA. A pesar de los rumores que corren por algunas BBS, y como muy bien avisa el creador François Lionet en varios mensajes, el AMOS no las admite y las "versiones" que sí las admiten son versiones modificadas por "hackers" para forzar a que las admita.

Otra desventaja de no aprovechar el sistema operativo, es que todas las novedades que se van incorporando al sistema tampoco pueden ser aprovechadas. Por ejemplo, si un usuario dispone del duplicador de discos Xpk, el cual permite duplicar la capacidad del disco duro, el AMOS no puede acceder al disco duro (o partición) donde esté siendo aplicado el Xpk, con la consiguiente molestia de no poder leer todos los ficheros que deseemos. El creador del AMOS ha prometido que, en la próxima versión de éste, ya estará integrado plenamente en el sistema operativo y podremos disfrutar del entorno del Workbench, también desde AMOS, así como crear aplicaciones con un aspecto más profesional.

De todas maneras, estas carencias pueden ser subsanadas ampliando las posibilidades iniciales de AMOS. Sí, se le pueden añadir nuevos módulos, que, poco a poco, van mejorando las características del lenguaje. Como ejemplo sirva el módulo de manejo de gráficos vectoriales voodoo3d, el cual se puede incorporar a cualquier AMOS versión 1.34 o superior, y por supuesto AMOSPRO. Con este módulo se pueden crear "demos" vectoriales como algunas de las que solemos ver en nuestro ordenador de vez en cuando, y que asombran, sobre todo, a usuarios de otros equipos. Además de éste, actualmente están disponibles los siguientes:

□ Idos: Mejora el acceso a ficheros y matrices del AMOS e incorpora comandos nuevos para manejo de AREXX. No es necesario en el AMOSPRO.

□ Iser: Permite acceder al puerto serie y aprovecharlo desde AMOS. Tampoco es necesario en el AMOSPRO.

□ craft2: Multiplica espectacularmente la velocidad del AMOS y añade nuevos comandos gráficos. Ideal para hacer demos y juegos. No es compatible con AMOSPRO. Es imprescindible para programadores de juegos y demos.

□ compiler: Este módulo es necesario para poder hacer ejecutables nuestros programas. Con éste, ganamos velocidad y nos aseguramos de que nadie eche vistazos indiscretos al listado de nuestro programa. Hay versiones para AMOS y AMOSPRO.

□ tome: Muy útil para programadores de juegos. Nos ofrece comandos que facilitan el manejo de mapeados de juegos. No es compatible con AMOSPRO.



Rafael Romero nos introduce a un lenguaje de programación sencillo, potente, y utilizado por muchos, que desplazó rápidamente al Amiga Basic.

A

M

O

S

□ tracker: Con él podemos leer módulos compatibles con noisetracker, protracker, etc., y, por supuesto, escucharlos. No es necesario en AMOSPRO.

Ahora, voy a comentar las variables y los tipos disponibles. Si usted ya conoce el lenguaje basic de "otros" sistemas, lo siguiente le sonará.

Un nombre de variable válido es aquel que no coincide con ninguna palabra clave de AMOS. Cuando estamos editando un listado de programa, el editor chequea automáticamente la línea que hemos introducido, capitaliza las instrucciones de AMOS y pone en mayúsculas las variables. Esto va muy bien a la hora de buscar instrucciones y variables a ojo. Por ejemplo, son nombres de variables válidos los siguientes: contador, índice, cuenta, libro, etc. Nosotros no tenemos que preocuparnos de tener que escribir en mayúsculas o capitalizar ya que el editor lo hace automáticamente.

Los tipos de datos disponibles en AMOS son los siguientes:

○ **BOOLEANO**: Sólo permite dos valores: TRUE (equivalente numérico -1) y FALSE (equivalente numérico 0). Su declaración puede ser cualquier letra o palabra que no coincida con una palabra clave de AMOS. A las variables de tipo booleano se les puede asignar los valores lógicos True y False, otras variables de tipo booleano o el resultado de operaciones booleanas.

Ejemplo:

```
A=True : Rem la variable A toma por valor cierto
B=A
C=(10>20) : Rem En este caso C toma valor False
Print A : Rem imprimimos el valor de A en pantalla
Resultado-> -1
```

Nota: Como las variables booleanas tienen un valor numérico que conocemos podemos aprovechar esto para hacer expresiones matemáticas que dependan de un valor booleano. Es perfectamente válido lo siguiente

```
A=True
Print A*180
Resultado-> -180
```

aunque no recomiendo su uso.

○ **NUMÉRICO** : Rango en precisión simple 1 E-14 a 1 E+15. Rango en precisión doble 1 E-307 a 1 E+308. No se puede trabajar con los dos tipos de precisión a la vez, sólo con uno. Hay que declarar la precisión al comienzo del programa, siendo por defecto precisión simple.

Además del tipo de precisión, hay que tener en cuenta que se dispone de dos tipos de datos numéricos: enteros y reales. Los reales son más precisos, pues admiten decimales pero por contra son más lentos al efectuar cálculos.

La notación para variables enteras es la misma que para las variables booleanas (de hecho, las variables booleanas son un caso especial de este tipo de variables) mientras que, para las variables reales, la declaración es la siguiente

```
nombre_variable# = 1.0
```

donde nombre_variable es cualquier secuencia de caracteres que no corresponda con una palabra clave del lenguaje, el símbolo # (SHIFT+3) indica variable numérica real y el valor asignado es otra variable de tipo real o un número real con decimales, aunque los decimales sean 0. ¡Atención! Si no se pone este símbolo, #, después del nombre de la variable, el intérprete no dará ningún error, tratando a la variable como una variable de tipo entero.

Ejemplos:

```
Rem Ejemplos números enteros
B=1 : Rem Asignamos el valor 1 a la variable B
B=2+1 : Rem Asignamos a B el resultado de la suma
B=A+C : Rem Asignamos a B el resultado de sumar A y C
```

```
Rem Ejemplos números reales
B#=1.0 : Rem Atención al símbolo y a la asignación
con decimales
B#=2.0+1.0
B#=A#+C#
```

También podemos "mezclar" los dos tipos de variables numéricas si hace falta, pero hay que tener en cuenta a qué tipo de variable hacemos la asignación.

```
nombre_variable_real = nombre_variable_entera#
```

```
nombre_variable_entera = nombre_variable_real#
```

¡Atención! en el segundo caso al hacer la asignación el valor es redondeado.

Ejemplos:

```
A=1 : Rem A variable entera.
B#=1.5 : Rem B# variable real.
Print A+B# : Rem Si no asignamos el valor, AMOS siempre
devuelve un valor real.
Resultado-> 2.5
```

```
C=A+B# : Rem C variable entera, al hacer la suma se redondea.
Print C
Resultado-> 2
```

```
C#=A+B#
Print C#
Resultado-> 2.5
```

Las operaciones que admiten las variables enteras y reales son las mismas que admiten números. Son la suma, resta, división, etc., además de las funciones trigonométricas.

Nota: Podemos tener dos variables que se llamen igual pero sean de diferente tipo. En el ejemplo anterior la variable entera C es diferente de la variable real C#.

○ **CADENA** : Permite almacenar hasta 65536 caracteres en una variable. Las variables de tipo cadena (o string) se declaran de la siguiente forma

```
nombre_variable$ = "texto"
```

donde nombre_variable puede ser cualquier palabra que no coincida con una palabra clave de AMOS. El símbolo \$ (SHIFT+4) indica que la variable es de tipo cadena. A las variables de cadena se les puede asignar una cadena, que es un texto entre comillas, u otra variable de cadena.

Ejemplos:

```
A$="AMOS v. 1.34"
B$="PRO"
C$=A$
Print C$
Resultado-> AMOS
```

Nota: Las cadenas admiten comparaciones y la función suma, además de muchas funciones especiales sólo para cadenas. La suma, en cadenas, se entiende como una concatenación.

Ejemplo:

```
A$="AMOS"
B$="PRO"
C$=A$+B$
Print C$
Resultado-> AMOSPRO
```

Podemos transformar datos de tipo numérico en cadena y viceversa con las funciones Str\$ y Val.

○ **MATRIZ**(x1,x2,...,xn): Nos permite almacenar nuestros datos en una tabla del tipo definido. La tabla puede ser de una o más dimensiones y tiene que ser declarada al principio de nuestro programa. No se pueden definir tablas que contengan más de un tipo de datos. Las tablas pueden ser booleanas, numéricas o de cadenas. Antes de poder guardar datos en una matriz hay que declararla con el comando

```
Dim nombre_tabla(tamaño).
```

Los índices en una matriz van de 0 a tamaño, ambos inclusivos.

Ejemplos:

```
Dim TABLA(10)
TABLA(0)=1
TABLA(1)=4
TABLA(10)=14
Print TABLA(0),TABLA(1),TABLA(10)
Resultado-> 1 4 14
```

Estos son todos los tipos de datos que admite AMOS. Además, nos permite reservar zonas de memoria para poder guardar datos que nos interesen. Las zonas de memoria se pueden reservar temporalmente o para todo el tiempo que esté en marcha nuestro programa, pero esto lo explicaré en otra ocasión.

Aunque no permite definir tipos de datos, sí permite una cosa que es muy valorada hoy en día: la programación estructurada. Si no comprende a que me refiero, no se preocupe pues con el ejemplo le quedará claro. Le basta saber, de momento, que con la programación estructurada es bastante más fácil programar algoritmos complejos y largos.

Ahora, como lo prometido es deuda, paso a dar un ejemplo. Primero explico el programa en un lenguaje similar al pascal y luego doy la versión en AMOS.

Versión en lenguaje algorítmico:

Programa fondo_animado;

```
procedimiento Crear_Bobs()
borrar_pantalla
lapiz:=3
Para radio:=1 hasta 8
(*Dibujamos círculos de origen 8,8 y radio de 1 a 8 *)
Circle 8,8,radio
(*Cogemos una porción de pantalla y la guardamos *)
(* como objeto gráfico *)
Get Sprite 0,N,0,0 To 31,31
borrar_pantalla
finPara N
Para radio:=7 hasta 1 paso -1
Circle 8,8,N
Get Sprite 0,15-radio,0,0 To 31,31
borrar_pantalla
finPara N
fin Crear_Bobs;
```

```
procedimiento _AnimarPantalla()
A:=2;
Repetir
lapiz:=4; papel:=2;
(* Seleccionamos un tipo de relleno para los procedimientos. *)
(* El signo negativo indica que nos referimos a una imagen *)
(* del banco de objetos gráficos *)
relleno=-A;
(* Dibujamos un rectángulo *)
Bar 0,0 To 320,200;
Si a>15 entonces a:=2;
sino a:=a+1;
finSi
Hasta Pulsar_Botón_ratón=Cierto;
Borrar_banco_de_objetos_gráficos;
fin _AnimarPantalla;
```

(* Inicio del programa principal *)

```
Abrir_pantalla (0,320,200,16,Lowres);
Cambiar_Atributos_Pantalla;
Crear_Bobs();
_AnimarPantalla()
```

fin fondo_animado.

Ahora la versión en AMOS:

Rem --- Cortar desde aquí ---

```
Rem Coger el programa y grabarlo como ejemplo1.asc
Rem Usar la opción "Merge Ascii" del menú de carga y luego "Run"
Rem Este programa crea un fondo animado.
Rem Los parpadeos es posible evitarlos, como ya se verá más adelante
Rem Programa principal
```

```
Rem Abrir pantalla
Screen Open 0,320,200,16,Lowres
Flash Off
Curs Off
Proc CREAR_BOBS
Proc _ANIMARPANTALLA
End
```

```
Procedure CREAR_BOBS
Cls 0 : Rem Borramos la pantalla
Ink 3 : Rem Seleccionamos color del lápiz gráfico
For N=1 To 8
Circle 8,8,N : Rem Dibujamos un círculo de radio N
Rem Guardamos una porción de pantalla en el banco de objetos
Get Sprite 0,N,0,0 To 31,31
Cls 0
Next N
For N=7 To 1 Step -1
Circle 8,8,N : Rem Dibujamos un círculo de radio N
Rem Guardamos una porción de pantalla en el banco de objetos
Get Sprite 0,15-N,0,0 To 31,31
Cls 0
Next N
End Proc
```

Procedure _ANIMARPANTALLA

```
Rem Imprimimos una línea de texto, centrada, en pantalla
Centre "Para comenzar pulsa una tecla."
Rem Bajamos una línea
Print Cdown$
Centre "Para acabar pulsa un botón del ratón."
Rem Esperamos la pulsación de cualquier tecla
Wait Key
Rem Escondemos el gráfico del cursor del ratón
Hide
A=2
Repeat
Ink 4,2
Rem Escogemos tipo de relleno para dibujar
Set Pattern -A
Rem Dibujamos un rectángulo
Bar 0,0 To 320,200
Rem Equivalente a lo siguiente
Rem IF A>15 THEN A=2
Rem ELSE A=A+1
Rem ENDIF
Add A,1,2 To 15
Until Mouse Click
Rem Borramos el banco de objetos
Erase 1
Rem Hacemos aparecer el cursor del ratón
Show
```

End Proc

Rem --- Acaba aquí ---

Bueno, aquí acaba esta pequeña introducción. Si le ha parecido interesante el AMOS siga leyendo los siguientes capítulos. ●

METAL SOFT

SABEMOS QUE OS PUEDEN OFRECER D.P. POR MENOS, PERO NOSOTROS OFRECEMOS CALIDAD Y NOVEDAD A UN PRECIO RAZONABLE.

TE OFRECEMOS:

- * UTILIDADES
- * CLIPARTS
- * FONTS
- * PICS
- * MUSICA
- * JUEGOS
- * DEMOS
- * ANIMACIONES
- * PROGRAMAS X
- * NCS
- * AGA
- * IFF
- * IMG
- * POSTSCRIPT
- * COMPUGRAPHIC
- * 24 BITS
- * AGA
- * ECS
- * MODULOS-HIGH QUALITY
- * SAMPLES HIGH-QUALITY
- * SCORE'S Y INSTRUMENTOS PARA SONIX
- * LOS MAS ACTUALES
- * ESPECTACULARES
- * ANIM5
- * ANIM7

COLECCIONES:

DG, EXCLUSIVA (1-64)
FRED FISH (1-1000)
AMIGALIBDISK (1-760)
ASSASINS GAMES (1-141)
ETC.

DISPONEMOS DE 6 GIGAS DE INFORMACION, NO OBSTANTE SI NO VES REFLEJADO LO QUE BUSCAS NO DUDES EN CONSULTARNOS.

+ DE 15 DISCOS = 1 DE REGALO

+ DE 40 DISCOS = Nº DE SORTEO PARA UNA AMIGA CD 32BITS

PREGUNTA POR NUESTRO
SERVICIO 24h.

CLUB METAL LIFE

- * BAZAR GRATUITO
 - * CUOTA MINIMA
 - * SORTEOS
- POR INSCRIPCION BOLETO DE SORTEO DE CD 32BITS

ADQUIERE
NUESTRO
CATALOGO
GRATUITO

APDO. CORREOS 873-08220 TERRASSA-BARCELONA
TEL: (93) 783.13.31

EL EQUIPO DE AMIGA.INFO

¡SÓLO ELLOS LO HACEN POSIBLE!

Muchos de vosotros os habreis preguntado ¿Quiénes son estos de AlnFo? ¿Dónde están? ¿Cómo son?

La mayoría de nosotros somos universitarios y estudiamos en las Facultades de Informática y Telecomunicaciones de la Universidad Politécnica de Cataluña en Barcelona. Al estudiar en el mismo centro, era inevitable que algunos de nosotros nos conociéramos gracias al Amiga y formáramos una especie de 'Club del Amiga' donde todo el que quisiera podía participar. Algunos de los que pertenecíamos a este club estábamos bastante disgustados de la situación actual del Amiga en España y sobre todo de la poca información que poseíamos sobre el ordenador, por no decir que la única revista de Amiga que existía en castellano nos tenía ya más que hartos de tanta desinformación y poca

seriedad. Había que hacer algo. Alguien propuso hacer una revista y poco a poco fuimos formando un equipo de personas interesadas en esta idea...Amiga.InFo se puso en marcha.

Ahora ya han pasado cinco meses y durante este largo período hemos conseguido que muchas empresas se interesen por nosotros y presten su apoyo a este proyecto. Hemos tenido muchas dificultades en el tema de impresión y montaje con Amiga, pero una vez solucionados podeis comprobar el resultado en

estas páginas. ¿No está del todo mal verdad?

El pueblo desde donde se ha coordinado todo el proyecto se llama Molins de Rei, situado a las afueras de Barcelona. Es un pueblo pequeño, pero muy especial debido a que es el que mas amigos tiene por metro cuadrado y esto ha servido de caldo de cultivo para la revista... En cuanto al lugar de trabajo, un desván lleno de trastos viejos nos ha hecho las funciones de oficina, hasta que consigamos un local en condiciones...¡todos tenemos que empezar de alguna forma!



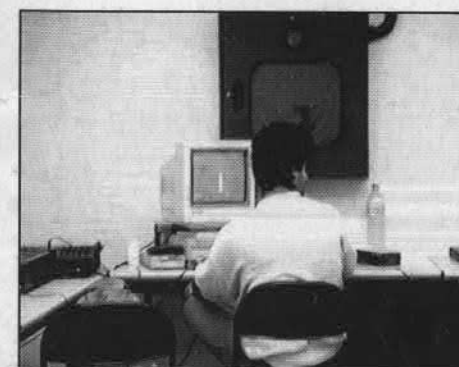
La última pregunta vosotros mismos podeis contestarla con estas fotografías donde aparecen la mayoría de nosotros (falta el que hace la foto, Francisco, que espero nos disculpe y algunos colaboradores que se han unido posteriormente...)

MATERIAL UTILIZADO

Amiga 3000
10 Megas de Ram
2 discos duros SCSI de 150 MB y 52 MB
2 disqueteras
Monitor MultiSync
Amiga 1200
Disco duro IDE Interno de 420 MB (3.5")
4 Mb de Ram (2 Megas en PCMCIA)
Fuente de alimentación de PC (importante, para que aguante jornadas de trabajo intenso)
Monitor MultiSync
2 disqueteras
Conexión en Red con Parnet
Impresoras: HP DeskJet 520, HP LaserJet 4P
Scanners: GT-6500 de HP
Digitalizadores de Vídeo:
VLAB 24 bits, Vidi Amiga 12
Removible SyQuest de 45 Megabytes
Renault Super 5 GTL.
Café, Coca Cola de 2 litros y patatas fritas.
Teléfono y Modem HST Courier V.FC 28.800 baud

SOFTWARE UTILIZADO

Professional PAGE 4.0 A
ADPro 2.5
ImageFX 1.5
DirOpus 4.11
Parnet 2.0
Quarterback 6.0
Postview 1.86
LHA 1.38
QuickGrab AGA 1.0



COLABORADORES

En AMIGA.InFo estamos interesados en publicar artículos y colaboraciones de nuestros lectores. Los requisitos para ser articulista de esta publicación son: saber redactar con claridad y dominar alguno o varios de los siguientes temas: Programación de Demos, Lenguaje Ensamblador, Secuenciadores MIDI, Programas 3D, EXX y Comandos del AmigaDOS. También podemos aceptar cualquier otro tema relacionado con Amiga. ¡Tú escoges!

Los trabajos que sean aceptados y publicados serán remunerados en función de su calidad. Y si estás interesado, pasarás a formar parte de nuestro equipo de redactores.

Las personas interesadas pueden escribir a la siguiente dirección adjuntando el trabajo realizado y sus datos personales con teléfono de contacto.

AMIGA.InFo (Referencia: Colaboradores)
APDO. de Correos 166
08750 Molins de Rei
Barcelona

PUBLICIDAD DISTRIBUIDORES

Si usted es un distribuidor de AMIGA y quiere insertar su publicidad en AMIGA.InFo, puede ponerse en contacto con nosotros enviando un FAX al número (93) 401 70 50 a la atención de AMIGA.InFo

Tienen Vdes. la posibilidad de utilizar para su publicidad hasta una página DIN A4 completa de **GRATUITA**. Si no desean este tamaño, pueden utilizar por ejemplo media página o un cuarto, según sus necesidades. Pueden utilizar gráficos que acompañen al texto. Si quiere insertar más de un DIN A4 consulte nuestras tarifas.

Envíe su publicidad a nuestro apartado de correos y de forma URGENTE, preferiblemente en FOTOLITO DIN A4 preparado ya para la imprenta. ATENCION: A la hora de montar su publicidad colocando textos y gráficos, tengan en cuenta que la revista se imprime en una sola tinta, con tramados y escala de grises. Por lo tanto, NO hagan separación de colores, avisen a la fotomecánica de que el documento es en una sola tinta.

AVANCE DEL PRÓXIMO NÚMERO



**¡ESPECIAL
NAVIDADES!**
con más de
100 páginas y
otros 2 discos.

- Además de las secciones incluídas en este número, publicaremos nuevas secciones con los siguientes nombres:
- ☐ **ESPECIAL HARDWARE Y SOFTWARE**
Con un reportaje de los productos más destacados del mercado nacional.
 - ☐ **PRIMEROS PASOS CON AMIGA**
Explicaciones, paso a paso, para los nuevos usuarios de Amiga.
 - ☐ **INICIACIÓN A LA PROGRAMACIÓN**
Destinado a los que todavía no se defienden bien con la programación.
 - ☐ **CURSO DE LENGUAJE ENSAMBLADOR**
Iniciaremos la publicación de un curso de programación en Ensamblador.
 - ☐ **DOMINIO PÚBLICO**
Muestra de las últimas novedades en programas de Dominio Público.
 - ☐ **ENTREVISTA AL DISTRIBUIDOR DEL MES**
 - ☐ **Y MUCHO MÁS... ¡RESERVA TU EJEMPLAR! ¡YA!**