



Amiga-Insider

**Die Intuition-Bibliothek
macht den Amiga zu dem, was er ist:
ein Computer mit fantastischer
Benutzeroberfläche.**

**Es steckt jedoch noch mehr in
dieser Bibliothek, was für das
Programmieren hochinteressant ist.**

Nachdem wir im ersten Teil des Kurses Exec unter die Lupe genommen haben, befassen wir uns nun mit Intuition. Dazu ist es notwendig, sich ein paar grundsätzliche Gedanken zum Sinn und zur Notwendigkeit einer Einrichtung wie Intuition zu machen: Wir wissen aus dem ersten Kursteil, daß das Multitasking eine saubere Zusammenarbeit der einzelnen Teile des Betriebssystems erfordert. Besonders die nur einmal vorhandenen Quellen (Resources: Devices oder Hardware) müssen sorgfältig verwaltet werden. Sie werden den Programmen (Tasks) bei Bedarf nach festen Regeln zugeteilt.

So stellt der Amiga Diskettenaufwerke, Drucker, die Tastatur und die Bildschirmausgabe jeweils dem Task zur Verfügung, den diese gerade benötigt. Verlangen mehrere Tasks gleichzeitig dieselbe Quelle, wird mit Hilfe der Priorität entschieden, welcher den Vorrang erhält. Bis zur Freigabe behält dann der Task die Kontrolle über diese Komponente, andere Tasks warten. So sinnvoll dieses Verhalten für Drucker oder Diskettenaufwerke sein mag, es ist offensichtlich, daß es für die Bildschirmausgabe und die Tastatur- und Mauseingabe nicht geeignet ist. Denn diese beiden Bereiche sind die wohl am häufigsten benutzten Teile des Systems. Müßte der Programmierer jedesmal die Tastatur oder den Bildschirm anfordern und wieder freigeben, wäre die Lust am Multitasking bald vorüber. Viel einfacher ist es, wenn man jedem Programm seinen »privaten« Ausgabebereich für Text und Grafik bereitstellt. Die Eingaben werden automatisch dem jeweils »aktiven« Programm zur Verfügung gestellt. Eben gerade diese beiden Aufgaben erledigt Intuition, und zwar ebenso flexibel wie effektiv. Auch wenn das bei erstmaligem Betrachten anders erscheinen mag.

Wenn wir von Intuition sprechen, meinen wir meistens die Screens, Windows, Requester etc., also die Benutzeroberfläche des Amiga. Die Intuition-Li-

brary enthält Informationen und Funktionen zur Verwaltung dieser Oberflächen-Objekte. Außerdem gibt es noch das Nachrichtensystem von Intuition, mit dem wir Nachrichten von oder über solche Objekte erhalten. Wir wollen den Aufbau und die Bestandteile dieses Systems ordnen und erläutern. Bevor wir zur Intuition-Library und dem Nachrichtensystem kommen, erklären wir die einzelnen Bestandteile der Oberfläche und ihren Zweck.

Die größte und allen anderen Objekten übergeordnete Einheit ist der Screen. Ein Screen ist ein eigener Grafikausgabebereich, der wie ein separater Bildschirm behandelt werden kann (engl. screen = Bildschirm). Ein Screen entspricht etwa einer Pinwand, auf die man direkt Grafik oder Text ausgeben kann. Wenn verschiedene Programme oder Programmteile auf dieser einen Fläche alle ihre Ausgaben machen würden, wäre das Chaos bald unübersehbar. Man müßte alle Ausgaben zwischenzeitlich löschen, womit sie verloren wären. Die Lösung für dieses Pro-

bleem sind die Fenster (Windows). Auf jedem Screen können fast beliebig viele Fenster geöffnet werden. Diese Fenster stellen voneinander unabhängige Ausgabebereiche auf einem Screen zur Verfügung. Sie erlauben aber neben erweiterten Grafikmöglichkeiten auch Eingaben. Wenn wir uns den Screen als eine Art Pinwand vorstellen, entspricht ein Window einem Blatt Papier auf einer solchen Pinwand.

Für die Eingaben von Text, Ja/Nein-Entscheidungen oder ähnlichem wäre es aber mühevoll, jedesmal eine eigene Benutzeroberfläche zu programmieren. Diese Arbeit ersparen uns Kommunikationsfenster (Requester), eine Art Eingabe-Formulare. Auf diesem Formular können dann Auswahl und Eingaben mit Hilfe von Schaltern (Boolgadget), Editierfeldern (Stringgadget) und Schieberegler (Proportionalgadgets) gemacht werden. Diese Objekte sind unter dem Oberbegriff Gadget zusammengefaßt. Gadgets gibt es auch direkt in Fenstern und in Screens, sie sind also ein sehr universeller Bestandteil der Oberfläche. Eine begriffliche Zuordnung fällt bei Gadgets schwer, die im Wörterbuch enthaltenen Formulierungen »Dingsda« und »Apparatchen« machen dies deutlich.

Schließlich gibt es noch die Menüs, die ebenso wie Requester einem Fenster zugeordnet werden. Sie dienen zur Auswahl von Programmen oder Programmteilen. Sowohl in Menüeinträgen als auch in Gadgets können Grafiken oder Texte enthalten sein, die in einem einheitlichen Format abgelegt sind. Die Texte wiederum können in einem wählbaren Zeichensatz (Font) dargestellt werden. Die Grafiken (Images) sind als Bitmuster abgelegt.

Alle diese Objekte werden von Intuition verwaltet. Die Hierarchie der Elemente sehen Sie in Bild 1. Die Intuition-Library enthält Funktionen zur Erzeugung, Veränderung und Abfrage dieser Objekte. Welche Eigenschaften diese Objekte haben und wie wir sie nutzen können, wird nun im einzelnen beschrieben. Die Intuitionfunktionen, die sich dazu verwenden lassen, werden nur genannt. Wer darüber Genaueres erfahren will, sollte sich in entsprechender Fachliteratur (siehe Literaturverzeichnis) die einzelnen Beschreibungen der Funktionen und Parameter ansehen.

Es können, solange der Speicher reicht, beliebig viele Screens geöffnet werden. Auf jedem dieser Screens kann unabhängig von den anderen gearbeitet werden. Ein Screen wird durch die Screen-Struktur beschrieben. Dort sind Art, Größe, Auflösung, Farben, verwendeter Zeichensatz für Texte (Font), Titel und die zum Screen gehörenden Grafikstrukturen enthalten. Die Auflösung, Farbzahl und die Abmessungen werden vor dem Öffnen des Screens festgelegt. Mögliche horizontale Auflösungen sind ohne Flags (320 Punkte) oder HIRES (640 Punkte horizontal). Zusätzlich kann ein Screen mit doppelter Zeilenzahl (INTERLACE, 512 Punkte vertikal) und im HAM-Modus (Hold-And-Modify, 4096 Farben gleichzeitig) geöffnet werden. Die Anzahl

Teil 2

KURSÜBERSICHT

Amiga Insider verdeutlicht die Zusammenhänge und Abläufe im Betriebssystem des Amiga. Vorkenntnisse im Programmieren sind notwendig. Wenn Sie den Kurs verfolgen, machen wir sie zum Amiga Insider.

TEIL 1: Übersicht, Betriebssystem-Kern (EXEC), Nachrichtenübermittlung, Signale, Speicherverwaltung, Bibliotheken, Ausnahmestrukturen

TEIL 2: Intuition – Die Benutzeroberfläche, Intuition-Direktkommunikation, input.device.

TEIL 3: Geräte (Devices), input.device, keyboard.device, gameport.device, console.device, timer.device, serial.device, parallel.device, printer.device, audio.device, clipboard.device

TEIL 4: Grafik und Text auf dem Bildschirm – graphics.library, layers.library

TEIL 5: Zugriff auf die Diskette – dos.library, trackdisk.library

TEIL 6: Kochrezept – Wie baut man sich ein eigenes Device oder eine eigene Library?



der verfügbaren Farben richtet sich nach der Zahl der Ebenen (2 hoch Anzahl der Ebenen). Die maximale Anzahl der Ebenen (Planes) liegt gegenwärtig bei 6. Lediglich im HAM-Modus sind bis zu 4096 Farben möglich. Die Abmessungen eines Screens können größer sein als der auf dem Bildschirm sichtbare Bereich. Zur Zeit sind bis zu 1024 Bildpunkte in beiden Richtungen zulässig. Die Themen Auflösungen und Farben werden wir im Kursteil 4 (Grafik und Layers) ausführen. Zum Öffnen eines Screens muß eine NewScreen-Struktur initialisiert werden. Dort legt man die Auflösung, Anzahl der Ebenen, Farben, Maße und bei Bedarf Gadgets, Font und Titel fest. Der Typ eines selbstgeöffneten Screens ist immer CUSTOMSCREEN. Mit dem Aufruf der Intuition-Funktion OpenScreen(), die einen Zeiger auf diese NewScreen-Struktur als Parameter erhält, wird der Screen geöffnet, sofern das vorhandene Chip-RAM ausreicht. Denn ein Screen braucht ja für seinen ihm eigenen Grafikbereich entsprechenden Speicher. Bei sechs Ebenen sind das bis weit über 100 KByte, je nach Größe des Screens. Nach dem erfolgreichen Öffnen, erhält man einen Zeiger auf die zum geöffneten Screen gehörige, vom Sy-

Viele Strukturen

stem angelegte, Screen-Struktur. In dieser Struktur sind alle Informationen über den Screen und die in ihm enthaltenen Grafikbereiche und Fenster enthalten. Beispiele hierfür sind die Zeichenfarben, der Font, die Abmessungen etc. Diese Daten sind die Grundlage für alle weiteren Aktionen auf einem solchen Screen. So werden die Farben und Fonts für alle Fenster zunächst dieser Screen-Struktur entnommen. Auch stellen die Grenzen des Screens die Grenzen für alle anderen Objekte auf diesem Screen dar, etwa für Windows oder Grafikausgaben. Mit CloseScreen() kann ein Screen geschlossen werden. Der vom Screen belegte Speicher wird dann an das System zurückgegeben. Allerdings nur dieser. Auf einem

Screen befindliche Fenster verschwinden zwar optisch, werden aber nicht im Speicher gelöscht. Der von solchen Phantomfenstern belegte Speicher ist verloren, denn löschen darf man solche Fenster dann auch nicht mehr. Also bitte vor dem Schließen eines Screens diesen immer erst »aufräumen«, das heißt alle Fenster löschen. Ob sich auf einem Screen noch Fenster befinden, läßt sich leicht feststellen, denn in der Screen-Struktur findet sich ein Zeiger auf das erste Fenster. Existiert kein Fenster auf dem Screen, ist der Zeiger NULL.

Jeder Screen enthält an seiner Oberseite einen Balken zum Verschieben und zwei Gadgets zum Vorholen beziehungsweise Zurücklegen. Dies kann auch durch Aufruf der Funktionen MoveScreen(), ScreenToFront() und ScreenTo-

zum Verändern der Größe und zum Schließen. Welche der Gadgets dargestellt werden, läßt sich beim Öffnen eines Fensters durch entsprechende Flags festlegen. Neben den Standard-Gadgets lassen sich zusätzlich auch eigene in ein solches Fenster einfügen. Fenster können mit automatischem Refresh (SMARTREFRESH) oder ohne (SIMPLEREFRESH) definiert sein. Im ersten Fall wird der Fensterinhalt nach dem Überlappen durch andere Fenster oder Requester automatisch restauriert. Im anderen Fall muß dies durch das Programm erfolgen. Dazu ist es möglich, sich entsprechende Nachrichten (Messages) von Intuition geben zu lassen. Jedem Fenster sind die Grafikstrukturen RastPort und Layer zugeordnet. Zu ihnen erläutern wir nur soviel, wie an

Grafikspeicherbereich (Bitmap) in der NewWindow-Struktur übergeben. Neben der Ausgabe sind Fenster aber auch die Empfangsstationen für alle Arten von Nachrichten. Dazu enthalten sie eine spezielle Vorrichtung: IDCMP (Intuition Direct Communications Message Port). Dies ist ein MessagePort, wie wir ihn aus dem ersten Kursteil schon kennen. Er dient zum Empfang aller Arten von Nachrichten, die diesem Fenster zugeordnet sind. Zum genauen Funktionieren dieses Ports kommen wir noch. Für Windows gibt es eine Vielzahl von Funktionen, von denen die wichtigsten genannt werden:

Das Öffnen geschieht ähnlich den Screens durch Initialisierung einer NewWindow-Struktur und einem Aufruf von OpenWindow(). Zum Schließen dient entsprechend CloseWindow(). Analog heißen auch die Funktionen zum Vorholen und Zurücklegen von Fenstern WindowToFront() und WindowToBack(). Zum Verschieben, Ändern der Größe und der Abmessungsgrenzwerte dienen MoveWindow(), SizeWindow() und WindowLimits(). Die Titel können mit SetWindowTitle() geändert und die Ränder mit RefreshWindowFrame() neu gezeichnet werden. Der Empfang von Nachrichten wird in der NewWindow-Struktur, oder bei schon geöffneten Windows mit ModifyIDCMP() eingestellt.

□ Zu jedem Fenster kann bei Bedarf ein Menü installiert werden. Dieses Menü wird bei Drücken der rechten Maustaste sichtbar, und aus ihm können dann Menüeinträge ausgewählt werden. Ein Menü be-

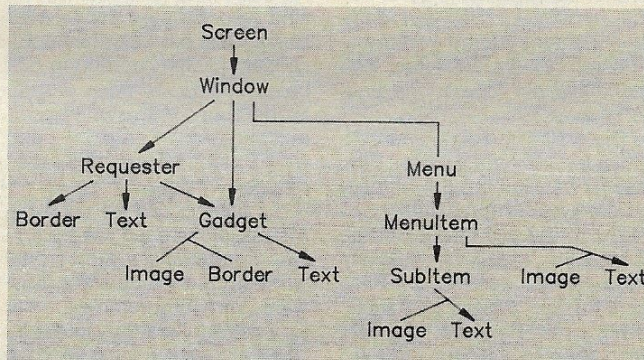


Bild 1. Die Hierarchie der Intuition-Elemente

Back() geschehen. Schließlich läßt sich das Anzeigen des Screen-titels mit ShowTitle() steuern. Zwar kann man auf Screens Text und Grafik ausgeben, aber dabei erfolgt keinerlei Begrenzung der Ausgabe (Clipping). Man muß also darauf achten, daß nicht über den Rand eines Screens gezeichnet wird, sonst wird »fremdes« Chip-RAM beschrieben.

□ Die Intuition-Fenster sind die Ausgabeeinheiten schlechthin auf dem Amiga. Windows verschiedener Typen können auf einem Screen geöffnet werden. Der Benutzer oder das Programm positionieren oder ändern die Größe beliebig. Auf ihnen kann Text oder Grafik ausgegeben oder auf das Eintreten bestimmter Nachrichten gewartet werden. Der Screen wird von allen diesen Aktionen nicht berührt, er bleibt »sauber«. Zu den Einrichtungen, die zu einem Fenster gehören, zählen die sofort ins Auge fallenden Gadgets zum Verschieben und Vorholen oder Zurücklegen, die wir ja vom Screen her schon kennen. Hinzu kommen die Gadgets

dieser Stelle unbedingt notwendig ist. Die Grafik beschäftigt uns im Kursteil 4. Der RastPort ist ebenso wie beim Screen der Ausgabebereich. Er verweist seinerseits auf die Bitmap mit den Zeigern auf die eigentlichen Zeichenebenen. Außerdem enthält er die für alle Grafikausgabefunktionen relevanten Daten wie aktuelle Zeichenfarbe oder Linien- und Füllmuster. Die Layer-Struktur enthält Daten, die für das Begrenzen der Ausgabe (Clipping) und den Neuaufbau des Inhalts (Refresh) erforderlich sind, wenn etwa der Fensterinhalt durch Überlappen von anderen Fenstern oder Requestern zerstört wird. Die Fenstergrenzen sind immer die äußerste Grenze für Ausgaben. Im Gegensatz zu Screens kann also bedenkenlos über die Ausmaße des Fensters gezeichnet werden, diese Ausgaben werden automatisch gekappt. Leider arbeitet dieses Clipping nicht immer fehlerfrei, es kann also zu Überraschungen kommen. Windows vom Typ SUPERBITMAP wird ein eigener

Fenster mit Menü

steht aus Menütiteln und -einträgen, gegebenenfalls auch noch aus Untereinträgen. Die Titel sind die Texte, die bei Drücken der rechten Maustaste sichtbar werden. Ihnen zugeordnet sind dann die Menüeinträge, die erscheinen, wenn die Maus auf den Selektionsbereich eines Titels bewegt wird. Jedem Eintrag kann dann noch eine Reihe von Untereinträgen zugeordnet werden, die bei Auswahl eines Eintrags sichtbar und selektierbar werden. Menüeinträge und Untereinträge werden intern durch eine MenuItem-Struktur verwaltet, die Titel durch eine sogenannte IntuiText-Struktur. Die MenuItem-Struktur enthält neben dem Zeiger auf einen Text (IntuiText) oder auf eine Grafik (Image) für einen Eintrag Informationen



darüber, ob er überhaupt selektierbar ist und wenn, wie er dargestellt werden soll, wenn er selektiert wird. Zusätzlich kann der Eintrag mit einer Tastenkombination als angewählt definiert und mit einem Häkchen versehen werden. Menüs müssen mit ihren Komponenten vom Programmierer komplett initialisiert werden, es gibt keine Funktionen dazu. Wer über eine Bibliothek oder ein Programm verfügt, die ihm diese Arbeit abnimmt, wird davon sicher regen Gebrauch machen. Lediglich das An- und Abmelden eines Menüs kann und muß mit den Funktionen `SetMenuStrip()` und `ClearMenuStrip()` erfolgen.

Bei Requestern handelt es sich um Formulare, die in einem Fenster dargestellt werden. Die Requester haben eigentlich nur die Funktion, als eine Art übergeordnete Instanz für die im Requester enthaltenen Gadgets zu dienen. Der Requester selbst ist nur ein Rechteck in einer bestimmten Farbe, das neben einem Text und einer Umrahmung, beliebige Gadgets enthalten kann. Ebenso wie einem Fenster sind ihm ein eigener Rastport und ein eigener Layer zugeordnet. Das ist notwendig, um den Fensterinhalt unbeschädigt zu lassen beziehungsweise nach Entfernen des Requesters wieder restaurieren zu können. Die in einem

Requester

Requester enthaltenen Gadgets werden beim Zeichnen des Requesters alle automatisch dargestellt und bei Löschen des Requesters wieder entfernt. Zum Darstellen eines Requesters muß die Funktion `Request()` aufgerufen werden. Das Entfernen kann vom Programm mit `EndRequest()` oder bei Betätigen eines »Ende«-Gadgets erfolgen. Der Aufbau eines Requesters ist ebenso schwierig wie der eines Menüs, wenn man von einer primitiven Abart dieser Spezies absieht, den »Auto«-Requestern. Erzeugt werden sie mit der Funktion `AutoRequest()`. Mehr als eine Ja/Nein-Abfrage in der linken oberen Ecke eines Screens, wie wir sie von den Sy-

stemmeldungen her kennen, ist damit nicht realisierbar. Für einen »normalen« Requester müssen die Strukturen des Requesters und aller Gadgets vor dessen Darstellung komplett initialisiert werden.

Gadgets sind in Screens, Windows und Requestern enthalten. Sie dienen der Auswahl oder Eingabe und ihre Gestaltungsmöglichkeiten sind fast unbegrenzt. Man unterscheidet drei Typen von Gadgets: Boolgadgets, Stringgadgets und Proportionalgadgets. Boolgadgets dienen der schlichten Auswahl, vergleichbar mit einem Feld zum Ankreuzen auf einem Papierformular. Stringgadgets dienen der Eingabe beliebiger Daten wie Texte oder Zahlen. Mit den Proportionalgadgets steht uns eine Art Schieberegler zur Verfügung. Mit ihm ist »analoge« Eingabe möglich, etwa zum Einstellen oder Anwählen bestimmter Bereiche von Daten. Für Gadgets gibt es natürlich auch eine eigene Datenstruktur. Diese enthält Abmessungen und Farben, Zeiger auf Text und Umrahmung, den Typ des Gadgets und ähnlich den Menüeinträgen Informationen darüber, ob und wie sich das Aussehen bei Auswahl verändern soll. Das Aussehen eines Gadgets läßt sich durch eine oder mehrere Umrahmungen (Border) oder eine Pixel-Grafik (Image) bestimmen. Darüber hinaus ist bei Stringgadgets und Proportionalgadgets noch eine spezielle Zusatzstruktur notwendig, die `StringInfo`- beziehungsweise `PropInfo`-Struktur mit weiteren Daten. Stringgadgets bieten für die Eingabe von Texten einen guten Komfort. So existieren Funktionen zum Löschen und Zurückholen sowie zum Durchscrollen des Textes im Eingabefeld. Für das Konstruieren und Einsetzen von Gadgets in Requestern gibt es zwar keine Funktionen, wohl aber für das Einfügen von Window-Gadgets. Gadgets können mit der Funktion `AddGadget()` in Windows eingefügt und mit `RemoveGadget()` entfernt werden. Zum erstmaligen oder erneuten Darstellen der Gadgets nach Veränderungen der Gadgetliste sind die Funktionen `RefreshGadgets()` oder `RefreshGList()` aufzurufen. Ferner können Gadgets mit `OnGadget()` und `OffGadget()` aktiviert und deaktiviert werden. Sie werden dann normal beziehungsweise gerastert (ghosted) dargestellt. Bild 2 zeigt wie der Aufbau aussehen kann.

In fast allen Objekten, die von Intuition verwaltet werden,

kommen an einer oder mehreren Stellen Texte vor. So etwa bei Screens und Windows die Titel, bei Requestern und Gadgets die Texte, bei Menüs und (Unter-)Einträgen die Bezeichnungen. Diese Texte können in festgelegten Zeichensätzen oder im Standard-Systemfont (dazu später) ausgegeben werden. Bei Screen- und Windowtiteln sowie den Kopfeinträgen der Menüs werden in den zugehörigen Strukturen einfache Textzeiger verwendet. Diese verweisen auf den darzustellenden Text. Bei ihrer Darstellung wird der Font des entsprechenden Screens verwendet. Bei allen anderen Objekten, wie bei Gadgets, Menüeinträgen und Requestertexten wird eine spezielle Textstruktur verwendet, die `IntuiText`-Struktur.

wir die im Speicher, oder auf Diskette stehenden Fonts erfragen. Wir erhalten eine Liste der verfügbaren Fonts. Im Speicher befindliche Fonts können mit `OpenFont()`, die auf Diskette enthaltenen mit `OpenDiskFont()` geöffnet und nutzbar gemacht werden. Diese Funktionen bekommen Text-Attribut-Strukturen übergeben und liefern einen Zeiger auf eine Font-Struktur. Die so erhaltenen Fonts können mit `SetFont()` einem Fenster- oder Screen-Rastport zugewiesen werden. Mit der Funktion `SetFontStyle()` ist es ferner möglich, den Text kursiv, fett oder unterstrichen darzustellen. In den `IntuiText`-Strukturen stehen stets Zeiger auf Text-Attribut-Strukturen, oder eben NULL, keinesfalls aber ein Zeiger auf einen Font.

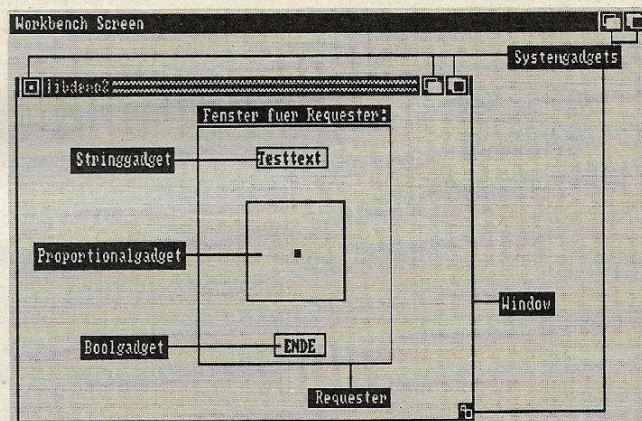


Bild 2. So können Elemente von Intuition aussehen

In ihr sind neben einem Zeiger auf den Text die Zeichen- und Hintergrundfarbe, der Darstellungsmodus, die Ausgabekoordinaten und ein Zeiger auf eine sogenannte Text-Attribut-Struktur (`TextAttr`) enthalten. Es kann also genau und unabhängig von anderen Einstellungen festgelegt werden, wo und wie dieser Text ausgegeben wird. Die Text-Attribut-Struktur beschreibt dabei einen Font, der nach Möglichkeit benutzt werden soll. Ist ein solcher Font nicht verfügbar, wird der Standardfont verwendet. Dieser Standardfont (meist Topaz 8) wird auch verwendet, wenn kein Font angegeben ist. Der Zeiger auf die Attribut-Struktur ist dann NULL. Wie kann man nun dafür sorgen, daß ein Font verfügbar ist? Es ist nicht ganz einfach, aber machbar:

Zunächst müssen wir unterscheiden zwischen Fonts, die sich bereits im Speicher befinden und dem System bekannt, sowie denen, die auf Diskette abgelegt sind. Mit Hilfe der Funktion `AvailFonts()` können

Sofern der angegebene Font verfügbar ist, wird er auch verwendet. Falls er nicht verfügbar oder nicht definiert ist, kommt ein Standardfont zur Verwendung. »Ein« Standardfont deshalb, weil sich die Entwickler hier nicht festgelegt haben – und dies auch nicht wollen. In Hinsicht auf zukünftige Bildschirmauflösungen ist dies auch sinnvoll. Wichtig ist, daß Sie die Abmessungen des Textes unter Verwendung des gewählten Fonts bei der Berechnung der Abmessungen von Gadgets und Requestern berücksichtigen. Mit der Funktion `IntuiTextLength()` kann die Länge eines Textes exakt ermittelt werden. Mit `SetFont()` schließen Sie einen geöffneten Font, egal um welche Art von Font es sich handelt. Dies sollte immer dann geschehen, wenn Sie den Font nicht mehr benötigen. Da es sich bei den Font-Routinen um Funktionen aus der Graphics-Library handelt, ist diese vorher zu öffnen. Siehe hierzu den ersten Teil des Kurses (Libraries). Bei Zugriffen auf Dis-

kettenfonts ist in jedem Fall zu- vor das Öffnen der Diskfont-Library notwendig. Nach Abschluß der Operationen mit Fonts sind diese Libraries, wie im ersten Kursteil beschrieben, wieder zu schließen.

□ Bei vielen der in Intuition verwendeten Strukturen sind Koordinaten für die Darstellung der Objekte enthalten. Diese Koordinaten beziehen sich jeweils auf ein übergeordnetes Objekt. So beziehen sich die Window-Koordinaten auf die linke obere Ecke des Screens, die Requesterkoordinaten auf die linke obere Ecke des Windows. Dabei ist zu beachten, daß der Ursprung in GIMMEZERO-Windows innerhalb der Ränder liegt, bei anderen Windows am äußeren Rand. Für Koordinaten von Gadgets in Windows gilt dasselbe, bei Requester-Gadgets ist allerdings der Requester das übergeordnete Objekt. Bei Gadgets besteht aber immer die Möglichkeit, den Bezugspunkt zu wählen: entweder von links oder rechts oder von oben oder unten. Dies wird durch das Setzen oder Löschen spezieller Flags in der Gadget-Struktur gesteuert. Auch in Menüs gibt es diese Bezugspunk-

te. Für ein Menü ist zunächst der Screen das Bezugsobjekt. Die Koordinaten für Menüeinträge beziehen sich auf das Menü, die Untereinträge auf den Eintrag. Bei Gadgets, Menüs und ihren Einträgen ist zwischen ihren optischen Abmessungen und ihrem Selektionsbereich zu unterscheiden. Die Texte von Gadgets werden bis zum Rand des Requesters angezeigt, unabhängig vom Bereich, in dem ein Gadget angewählt wird. Die Texte von Menüs und ihren Einträgen erscheinen bis zum Rand der umschließenden Box, auch hier kann der Wahlbereich ein anderer sein. Es ist bei Gadgets und Menüeinträgen ohne weiteres möglich, daß der Anwahlbereich ganz woanders liegt als der Text oder die Grafik dazu. Die in den Strukturen definierten Abmessungen geben stets den Selektionsbereich an.

Der Amiga besitzt einen speziellen Mechanismus zur Übertragung von Nachrichten. Wie wir bereits zu Beginn ausgeführt haben, dient der Nachrichtenmechanismus von Intuition dazu, Informationen über die Aktivitäten des Benutzers (oder auch eines Programms)

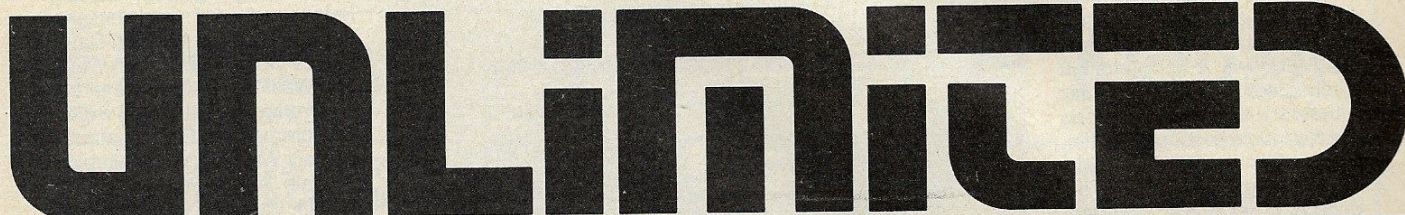
an die richtige Stelle weiterzuleiten. Screens können durchaus von mehreren Programmen gleichzeitig genutzt werden, denken wir nur an den Workbench-Screen. Sie eignen sich also nicht als Empfangsstationen. Windows hingegen sind nur für ein Programm zugänglich, sie sind also geeignet zur Unterbringung eines Empfängers (Messageport). Dieser Empfänger ist der bereits angesprochene IDCMP-Messageport. Alle Nachrichten, die sich auf dieses Window beziehen,

werden von einem Interrupt-Handler gelesen und ausgewertet. Einen solchen Handler besitzt auch Intuition. Intuition generiert aus den Input-Events die zugelassenen Intuition-Messages und sendet sie an den zuständigen Messageport des entsprechenden Windows (Bild 3). Die übrigen Ereignisse werden einfach »weggeworfen«. Jedes Programm kann einen solchen Handler installieren, allerdings ist die Priorität dieses Handlers entscheidend dafür, welcher davon zuerst drankommt. Intuition hat einen Handler mit hoher Priorität (50). Will man also die vom Input-Device erzeugten Events vor Intuition auslesen, muß man mindestens eine Priorität von 51 für den eigenen Handler angeben. Dann kann man die vom Input-Device erzeugten Events verändern oder löschen. Der Handler des Console-Device hat übrigens eine niedrigere Priorität.

Aber zurück zu Intuition. Eine Intuition-Nachricht besteht zum Teil aus einer gewöhnlichen Exec-Message-Struktur. Daran anschließend sind Informationen über die Art des Ereignisses, ein Codefeld, sowie die

Nachrichten

werden in diesem Port aufgezeichnet. Wie sieht nun eine Intuition-Nachricht aus und wie kommt sie zustande? Wenn wir eine Taste drücken, sendet der Tastaturprozessor eine entsprechende Information an das Keyboard-Device. Dieses wiederum meldet das an das Input-Device, das ja alle Informationen über Eingaben von Tastatur oder Maus sammelt. Das Input-Device macht aus diesen Informationen Ereignisse in einem einheitlichen Format. Diese in der InputEvent-Struktur beschriebenen Ereignisse kön-



Neu! Amiga Enhancer Kit 1.3 Workbench, Kickstart, Extras & Handbuch 79,- DM, Kickstart 1.3 ROM 59,- DM, Speicherchips 1024x1 100ns je 59,- DM,

Spiele		Animation		Disketten		Grafik		Simulation		Unlimited	
AARGH	55	AEGIS Animator & Images	219	3,5 Zoll 2DD No Name 10er	23	AEGIS Draw	169	Flightsimulator I deutsch	98	Unbegrenzt sind unsere Angebote zwar	
ALIEN FIRES	64	AEGIS Light & Camera Action!	149	5,25 Zoll 2DD No Name 10er	9	AEGIS Draw 2000	399	Galileo 2.0 Planetarium	89	nicht, doch bemühen wir uns, Ihnen	
ARAZOK'S TOMB	50	AEGIS Modeller 3-D	149	5,25 Zoll 2DD No Name	39	AEGIS Images	59	Jet	79	immer die günstigsten, und bestien	
ARFANOID	49	AEGIS Videoscape 3D 2.0 deutsch	289	Diskettenbox 5,25 DB-100 1-reihig	19	AEGIS Impact	115	Leader Board Golf	59	Produkte aus einem Angebot von über	
AUTODUEL	69	AEGIS Videotitle 1.1 deutsch	195	Diskettenreinigungssatz 3 1/2	12	Butterworth	59	Leader Board Tournament Dendisk	79	2000 Artikeln anzubieten. Alle mit deutsch	
BACKLASH	49	ANIMATE 3D	90	Diskettenreinigungssatz 5 1/4	15	Butcher 2.0 PAL deutsch	85	Scenery Disk #11	49	gekennzeichneten Programme sind aus-	
BAD CAT	25	APPRENTICE DISNEY 3D jr.	180	Diskettenreinstoff 3 1/2	12	Calligrapher	155	Scenery Disk #7	49	schließlich Originalprodukte der jeweiligen	
BALANCE OF POWER	59	Comicsetter deutsch	180	Diskettenreinstoff 5 1/4	19	DELUXE Paint II/Print deutsch	29	Scenery Disk Europe	49	deutschen Distributoren mit vollem Up-	
BALL RAIDER	29	DELUXE Productions	325	Diskettenreinstoff 5 1/4	29	DELUXE PhotoLab deutsch	199	Scenery Disk Japan	49	dateservice. Sie kaufen also keine	
BALLYHOO MYSTERY	69	DELUXE Video 1.2 deutsch	180	Diskettenreinstoff 5 1/4	29	DELUXE Seasons & Holidays	95			selbstgezeichneten Ventilen. Aus Kos-	
BARBARIAN	59	Pageflipper deutsch	50			EASYL 1000 Zeichentablett	649			tengründen haben wir keine Prospekte zu	
BATTLESIPS	51	Pageflipper plus F/X PAL	249			EASYL 2000 Zeichentablett	649			den angebotenen Artikeln.	
BONE CRUNCHER	49	Photom Video Cell Animator	249			Intro Cid	69				
BORROWED TIME	59	Scult 3-D	145			Printmaster plus	74				
BRAINSTORM	29	Scult 4-D	695								
CHAMPIONSHIP BASEBALL	59	Video Effects 3-D	349								
CHAMPIONSHIP BASKETBALL	59										
CHESSMASTER 2000	61										
COMPUTER BASEBALL	69										
COOGANS RUN	44										
DEEP SPACE	55										
DEJA VU	29										
DEMOLITION	29										
DETONATOR	39										
DIABLO	95										
EARL WEAVER BASEBALL	57										
ENCHANTER	30										
ENFORCER	30										
FINAL MISSION	57										
FINAL TRIP	23										
FUP-FLOP	26										
GALAXY FIGHT	44										
GARRISON	39										
GOLDEN PATH	39										
GOLDEN PYRAMID	39										
GRAND SLAM TENNIS	72										
GUILD OF THIEVES	69										
HACKER II	75										
HEX	29										
INTO THE EAGLES NEST	59										
JINXTER	60										
JUMP JET	39										
KAMPFGRUPPE	59										
KING OF CHICAGO	60										
KNIGHT ORC	65										
KWASIMODO	29										
LEATHER GODDESSES	60										
LEVATHAN	59										
LIVYANS IN SPACE	59										
LURKING HORROR	63										
MAGICIANS DUNGEONS	79										
MINDBREAKER	64										
MINDSHADOW	29										
MISSION ELEVATOR	45										
MICHAELIS	20										
MOONMIST	79										
OGRE	25										
PHALANX	25										
PLUTOS	47										
POWERPLAY	30										
REISENDE IM WIND 1+II	49										
RETURN TO ATLANTIS	49										
ROADWARS	49										
S.O.I.	89										
SECONDS OUT	89										
SHOOTING STAR	24										
Bücher		Drucker		Erweiterung		Kalkulation		Text		Zubehör	
Amiga User's Guide	39	Drucker A-500/2000 Centr.	15	Speicher 2/8 MByte A-2000	1398	Analyze 2.0	225	Pagesetter deutsch	198	Jitter Rid Filterscheibe	29
Das grosse Public Domain Buch #1	49	Einzelblattrechner NEC P2200	298	Speicher A-500.512KByte/Uhr	398	HAICALC	198	Pagesetter LaserScript	74	Abdeckhaube System & Monitor	24
Das grosse Public Domain Buch #2	49	Einzelblattrechner Star LC-10	245	TV-HF Modulator A-500/2000	59	Logistix 1.25 deutsch	299	Pagesetter Upgrade deutsch	30	Abdeckhaube Tastatur	12
Das Anwenderhandbuch Amiga	29	Einzelblattrechner Star NB 24-10	295	XT Erweiterung für A-2000	895			Scilable	189	AMIGA Scart Kabel 2 mtr.	24
Developers Reference Guide	49	Hewlett Packard Desk Jet deutsch	1998	AT Erweiterung für A-2000	2295			Vizaverte deutsch 2.0	199	AMIGA Originalmaus	180
DOS Handbuch Amiga	29	Hewlett Packard Desk Jet deutsch	1998	MPS 1230 9 Nadeln ser. & par.	498			Zuma Fonts I	199	Control Center Amiga 500	14
Handbuch NEC P6/P7 deutsch	98	MPS 1230 9 Nadeln ser. & par.	498	STAR LC 24/10 deutsch	639			Zuma Fonts II	199	Konzepthalter schwenkbar	14
		STAR LC 24/10 deutsch	639	STAR NB 24/10 deutsch	748			Zuma Fonts III	69	Mouse House Max grau	14
		STAR NB 24/10 deutsch	748	STAR NB 24/15 deutsch	1298					Mouse House Millie rosa	14
										Mouse Pad EXTRA 27 x 23 cm	14
										Multisync Monitorabbel	14
										Trackball	98
Deutsche Handbücher		Farbbänder		Festplatten		Lernen		Video		BESTELLSERVICE	
AEGIS AUDIOMASTER	29	Citizen 1200 sw.	14	CSM 20 MB SCSI I A-2000 kompl.	895	Funktion	79	Digi-View Gold PAL Digitizer	328	Rund um die Uhr 06121/543848	
AEGIS SONIX	39	MPS 803 sw.	18	GVP SCSI Hardcard 20 MB 28 msec.	1495	Länder dieser Erde	39	Flicker Filter (Non Interface)	999		
AEGIS VIDEOTITLE	39	MPS 1500 sw.	25	GVP SCSI Hardcard 40 MB 11 msec.	1995	Mathematik deutsch	149	Future Sound II	87	Wir liefern nur Originalprogramme zu knallhart kalkulierten Preisen.	
Balance of Power	29	NEC P6 plus sw.	12	GVP SCSI Hardcard 40 MB 11 msec.	2250			GENLOCK Comfo I A-2000	139		
Calligrapher	29	NEC P6 plus sw.	12	STAR LC 24/10 sw.	19			GENLOCK Comfo II A-2000	139	Bestellen Sie schriftlich oder unter obiger Telefonnummer. Lieferung	
Comicsetter	29	NEC P6 plus sw.	12	STAR NB 24/10 sw.	19			PAL Video Karte I A-2000	499		
Flugsimulator II	29	NEC P7 Color	99	STAR NB 24/15 sw.	25			Video Kamera Panasonic WV-1410	998	solange Vorrat reicht gegen Vorkasse (-10,- DM Porto) oder Nach-	
Kampfguppe	29	STAR LC 24/10 sw.	19					Videoobjektiv WV-1410 18mm	98		
		STAR LC 24/10 sw.	19							nahme (+10,- DM Porto). Mindestbestellwert 50,- DM. Preisänderun-	
		STAR NB 24/10 sw.	19								
		STAR NB 24/15 sw.	25							gen und Irrtümer vorbehalten.	
		STAR NB 24/15 sw.	25								
										UNLIMITED	
										M. Hottenbacher, Kehrstraße 23, 6200 Wiesbaden	



Adresse des Objektes enthalten, durch das die Nachricht zustande kam. Dieses Objekt kann ein Gadget, ein Menü-Eintrag oder auch das Window selbst sein. Weiterhin sind die Maus-Koordinaten relativ zum Fenster, die Systemzeit, zu der das Ereignis eingetreten ist und schließlich der Zeiger auf das Window enthalten. Im Programm kann man dann diese Nachrichten auslesen. Bei Menüs ist das Entschlüsseln der Information, welcher Eintrag ausgewählt wurde, mit Hilfe von speziellen Makro-Aufrufen möglich. Sie liefern uns dann als Resultat einen einfachen Index. Jede Nachricht läßt sich auf diese Weise eindeutig identifizieren.

Welche Nachrichten in einem Port aufgezeichnet werden, läßt sich einstellen. Hierzu dienen die IDCMP-Flags, die ja bereits beim Öffnen eines Windows entsprechend gesetzt werden. Dann wird beim Öffnen des Windows durch Intuition automatisch ein Messageport initialisiert und der Empfang der Nachrichten eingeschaltet. Nachrichten werden beispielsweise erzeugt, wenn man ein Window in der Größe verändert, ein Gadget in einem Window oder in einem Requirer auswählt oder losläßt, einen Menüpunkt auswählt, der Fensterinhalt neu gezeichnet werden muß oder eine bestimmte Zeit vergangen ist. Dabei ist Voraussetzung, daß die entsprechenden Flags für den Empfang dieser Nachrichten gesetzt sind.

Die IDCMPs

Bei einem geöffneten Fenster kann der Empfang durch Aufruf der Funktion `ModifyIDCMP()` jederzeit geändert werden. Dabei ist allerdings zu beachten, daß bei Abschalten des Empfangs durch Übergabe eines Wertes von `NULL` als `IDCMP`, auch der Messageport im Window samt aller Nachrichten gelöscht wird. Es empfiehlt sich daher, immer erst alle Nachrichten mit `GetMsg()` abzuholen und mit `ReplyMsg()` zurückzuschicken, wenn man sichergehen will, daß keine davon verlorengeht. Es ist auch möglich, die Nachrichten von mehreren Windows

auf einem Messageport zu empfangen. Dazu muß beim Öffnen aller weiteren Windows der IDCMP auf NULL gesetzt werden. Anschließend trägt man dann die Adresse eines Messageports als Userport in diese Windows ein und ruft dann `ModifyIDCMP()` auf. Da wir bei jeder Nachricht von Intuition auch die Adresse des Windows erhalten, von dem die Nachricht stammt, ergeben sich keine Probleme, festzustellen, von welchem Window die Nachricht stammt. Bei einem üppigen Nachrichtenempfang, etwa durch viele gesetzte Flags oder entsprechende Anwenderaktionen, sollte man dafür Sorge tragen, daß die Nachrichten

wird, ist meist nicht zu gebrauchen. Intuiticks werden auch nur bei aktiviertem Window gesendet, und nur wenn keine alten Intuiticks mehr anliegen. Sie sind für Zeitsteuerungen also wenig geeignet.

Die Intuition-Library stellt uns Funktionen zur Erzeugung und Verwaltung von verschiedenen Objekten zur Verfügung. Das Anzeigen und die Veränderung solcher Objekte werden aber nicht unmittelbar von anderen Programmen ausgeführt, sondern durch den Intuition-Task. So werden die Screens nach ihrer Reihenfolge sortiert, und zwar exakt so, wie sie hintereinander auf dem Bildschirm angeordnet sind. Der erste Screen

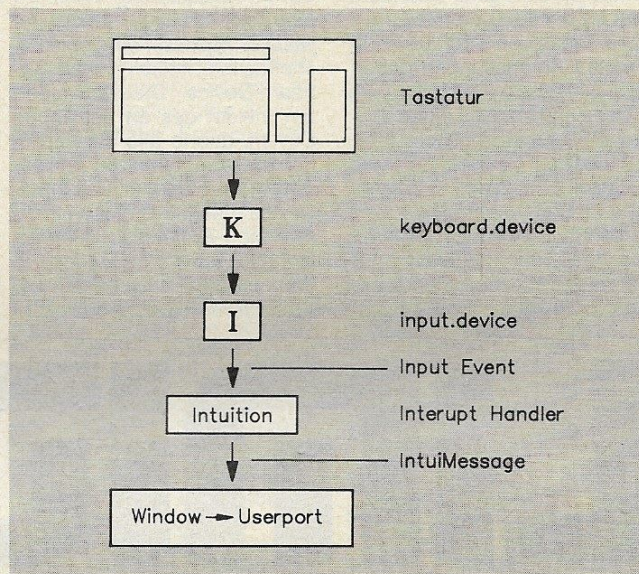


Bild 3. Der lange Weg eines Tastendrucks von der Tastatur zum IDCMP-Messageport eines Fensters

regelmäßig abgeholt und an Intuition zurückgesendet werden. Sonst kommt es zu einem Nachrichtenstau, was sich unter Umständen durch deutliche Geschwindigkeitsverluste beim Programmablauf bemerkbar macht. Hinzu kommt, daß jede Nachricht ja auch Speicher belegt. Daher sollten wirklich nur die Nachrichten erlaubt und die IDCMP-Flags dafür gesetzt werden, die tatsächlich vom Programm benötigt werden. Es macht keinen Sinn, auf Nachrichten von Window-Gadgets zu warten, die gar nicht existieren. Das gilt auch für Menüs und Requester. Man sollte nicht so verfahren, einfach alle Flags zu setzen und nicht benötigte Nachrichten wegzuworfen. Ganz besonders gut abwägen sollte man die Abfrage von Zeitgebernachrichten (INTUITICKS) und Mausebewegungen (MOUSEMOVE). Denn die Nachrichtenflut, die erzeugt

zeigt dann auf den nächsten, der wiederum auf den übernächsten etc. Auch die Windows auf einem Screen sind so miteinander verbunden, die Requester in einem Window, die Gadgets und die Menüs und Menü-Einträge. Es handelt sich also immer um eine Art Liste. Diese sind aber nicht zu vergleichen mit den Listen, die wir vom Exec her kennen. In der Intuition-Library sind neben, oder besser gesagt, vor den Sprungvektoren für die Funktionen die Informationen abgelegt, die Intuition für die Verwaltung dieser Objekte braucht. Ein Zeiger auf den ersten Screen und das aktive Window sind dort enthalten und können auch von anderen Programmen gelesen werden. Der erste Screen ist derjenige, der auf dem Bildschirm ganz oben liegt. Jeder Screen hat einen Zeiger auf den nächsten, dahinterliegenden. Die Screens

werden von Intuition ständig in dieser Ordnung gehalten. Windows haben zwar auch einen Zeiger auf das jeweils nächste, werden aber nicht sortiert. Screens haben einen Zeiger auf das erste Window, Windows einen auf das Menü, den ersten Requester und das erste Gadget, Requester einen auf den nächsten Requester und das erste Gadget, Gadgets einen auf den ersten Text und den ersten Border etc. Wir sehen, es herrscht Ordnung in Intuition. Zu dieser Ordnung wird man allerdings selbst auch von Intuition gezwungen, und wehe, man hält sich nicht daran. Bevor man auf Daten aus der Intuition-Library zugreift, ist der Zugriff für andere Tasks mit LockIbase() zu sperren. Ansonsten könnten sich diese Daten verändern, während wir sie lesen. Bei manchen Intuition-Funktionen ist zu beachten, daß das aufrufende Programm lediglich einen Auftrag an Intuition sendet, die beabsichtigte Operation auszuführen. Es kann vorkommen, daß diese Operation erst etwas später ausgeführt wird. Wenn man sichergehen will, sollte man auf die Beendigung lieber warten, indem man auf eine entsprechende Nachricht wartet. Dafür gibt es die Exec-Funktion Wait() und WaitPort(). Wait() wartet auf bestimmte Signale, egal woher sie kommen. Im Gegensatz dazu paßt WaitPort() auf beliebige Signale eines Ports auf. Der Port für Intuition-Nachrichten ist ja unser Window-Userport. Wenn wir uns zum Schluß noch klar machen, daß sämtliche Intuition-Objekte ja eine Art grafischer Objekte sind, die letztlich über die Graphics-Library und Layers-Library gezeichnet werden, erhalten wir erneut einen Eindruck von der engen Verzahnung des Amiga-Betriebssystems. Der Schluß, den man daraus ziehen muß, ist die Programmierung des Amiga unter Berücksichtigung der besonderen Verhältnisse. Alle Quellen und nötigen Speicherbereiche müssen vom System angefordert werden. Denn nur dann ist gewährleistet, daß das Multitasking und Ihre Programme weiter problemlos funktionieren.

Georg Zweschper

LITERATURVERZEICHNIS

LITERATURVERZEICHNIS
 Amiga-DOS-Handbuch; Amiga: Programmierpraxis
 Intuition, Amiga-Systemhandbuch
 Amiga-Programmierhandbuch; Markt & Technik Verlag
 Programmer's Guide to the Amiga, Amiga Programmer's Handbook; Sybex Verlag
 Amiga ROM Kernel Reference Manual: Series and Devices, Amiga Intuition Reference Manual; Addison-Wesley Verlag