

GoldDock

COLLABORATORS

	<i>TITLE :</i> GoldDock		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		November 24, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	GoldDock	1
1.1	Inhalt	1
1.2	Hinweis	1
1.3	COPYRIGHT	2
1.4	EINFÜHRUNG	3
1.5	INSTALLATION	3
1.6	API	4
1.7	Dock-Konfiguration	5
1.8	PROG	6
1.9	ICON	7
1.10	SOUND	7
1.11	DTOOL	8
1.12	DOCK	8
1.13	Tips	9
1.14	AREXX-Skripte	10
1.15	PRESETS	11
1.16	KEYLAYOUT	11
1.17	MENULAYOUT	13
1.18	PLÄNE	21
1.19	DANKSAGUNG	21
1.20	ADRESSE	22
1.21	Tastengebefehe	22
1.22	InputEvent-Klassen	23
1.23	Qualifier	24
1.24	Tastenkodes	25
1.25	rawkey Tastenkodes	25
1.26	rawmouse Tastenkodes	26
1.27	Beispiele für Tastengebefehe	26

Chapter 1

GoldDock

1.1 Inhalt

GoldDock V1.0

Copyright (C) April 1994 by Markus Aretz

HINWEIS !!!

Einführung

Copyright
Installation

Dock-Konfiguration

Tips

ARexx-Skripte
Presets

Pläne

Danksagung
Adresse

1.2 Hinweis

HINWEIS:
=====

Dieses Programm benötigt den hervorragenden Shareware-Editor GoldED von Dietmar Eilert sowie die ToolManager.library von Stefan Becker (FishDisk 872/873). Einige der mitgelieferten ARexx-Skripte nutzen die REXXReqTools.library, die im Paket der ReqTools.library enthalten ist.

Eine Demo-Version des GoldED ist erhältlich auf FishDisk 908/909. Diese

enthält jedoch meines Wissens noch nicht das Application Interface (API), mit dem Erweiterungen zum GoldED hinzugefügt werden können, wie z. B. dieses Dock-Interface oder aber ein SpellChecker, etc. Eine Demo-Version mit API-Interface ist auf dem AmiNet zu finden.

Die registrierte Vollversion, die ich jedem nur sehr empfehlen kann, ist bei

Dietmar Eilert

EMail: DIETMAR@TOMATE.OCHE.DE (Z-Netz)

erhältlich.

1.3 COPYRIGHT

COPYRIGHT UND ANDERE RECHTLICHE DINGE

Die in diesem Archiv enthaltenen Dateien sind

Copyright (C) 1994 Markus Aretz

"Am Anfang standen Schweiß und Mühe.

Also sprach der Programmierer:

Es werde Giftware! Und so war es."

(Zitat U. Röhm)

Da ich mit diesem Programm die Weiterentwicklung des GoldED unterstützen und anderen einen Anreiz geben möchte, diesen Editor zu benutzen, verzichte ich trotz eines nicht zu vernachlässigenden Zeitaufwandes, den ich bei der Programmierung und Erstellung der Docks hatte, auf irgendeine Registriergebühr.

Ich würde mich aber über Geschenke jeglicher Art von den Benutzern, die GoldDock regelmäßig verwenden, sehr freuen. Darunter können auch Icons oder Brushes sein, die Editor-Funktionen oder TeX-Aufrufe symbolisieren. Sollte es einem nicht möglich sein, mir irgendetwas zu meiner persönlichen Erbauung zukommen zu lassen, so möge er mir zumindest Vorschläge bzw. seine Meinung zu GoldDock schicken oder aber einen kleinen Betrag auf mein Konto überweisen (s. Adresse).

Es wird keine Garantie gegeben, daß die Programme in diesem Paket 100%ig zuverlässig sind. Sie benutzen dieses Programme auf eigene Gefahr. Der Autor kann auf keinen Fall für irgendwelche Schäden verantwortlich gemacht werden, die durch die Anwendung dieser Programme entstehen.

Das Paket ist "freely distributable", aber das Copyright liegt weiterhin bei Markus Aretz. Dies bedeutet, daß es von jedem kopiert werden darf, solange er nicht mehr als eine angemessene Kopiergebühr dafür verlangt. Diese Gebühr darf nicht höher sein als US \$5 oder 5 DM.

Dieses Limit gilt auch für deutsche Public-Domain Händler!!

Dieses Paket darf in Public-Domain Sammlungen aufgenommen werden,

insbesondere in Fred Fishs Amiga Disk Library (CD ROM Versionen dieser Sammlung eingeschlossen). Die Distributionsdatei darf in Mailboxsystemen oder auf FTP Servern abgelegt werden. Wenn Sie dieses Paket weitergeben wollen, dann müssen Sie die originale Distributionsdatei GoldDock.lha benutzen.

Weder die Programme noch der Quelltext (oder Teile davon) dürfen ohne eine schriftliche Genehmigung des Autors in kommerziellen Programmen benutzt werden.

Die Programme und der Quelltext (oder Teile davon) dürfen auf keinen Fall auf irgendeiner Maschine benutzt werden, die für die Forschung, Entwicklung, Konstruktion, Tests oder Produktion von Waffen oder anderen militärischen Gütern benutzt wird. Dies gilt natürlich auch für alle Maschinen, die für das Training von Personen in irgendeiner der obengenannten Tätigkeiten benutzt werden.

1.4 EINFÜHRUNG

EINFÜHRUNG

GoldDock entstand aus dem Wunsch, dass man in Zusammenarbeit mit GoldED eine frei definierbare Funktionsleiste – im weiteren nach NeXTstep bzw. ToolManager auch Dock genannt – zur Verfügung hat, die es ermöglicht Editorfunktionen, ARexx-Makros, WB-/CLI-Programme sowie Hotkeys einem beliebigen Piktogramm zuzuordnen.

Dies wurde realisiert unter Benutzung der toolmanager.library von Stefan Becker sowie unter Ausnutzung der API-Schnittstelle des GoldED.

Hiermit moechte ich zugleich andere Programmierer aufrufen, Applikationen für dieses Interface zu entwickeln bzw. jeden, der Ideen hat, was dort noch sinnvoll realisiert werden könnte, dies an Dietmar Eilert weiterzugeben.

Nachdem dies realisiert war, begann ich mehrere Docks zu entwerfen, insbesondere solche, die die Zusammenarbeit mit PasTeX von Georg Hessmann vereinfachen sollen.

So stehen jetzt Docks zur Verfügung, aus denen einfach griechische Buchstaben sowie mathematische Symbole in den TeX-Quelltext in der Art eines Formeleditors à la MicroSoft Word eingefügt werden können, eines zur Steuerung von PasTeX, eines für Blockfunktionen, eines zur Makrobearbeitung. Sie sind nicht alle vollständig belegt, sondern dienen vor allem zur Demonstration, was alles zu realisieren ist.

1.5 INSTALLATION

PROGRAMMINSTALLATION

Die Installation kann ganz einfach mittels des beigefügten Installer-Skripts durchgeführt werden. Im Konfig/API-Requester ist dann mit

Laden die gewünschte API-Konfiguration zu laden (Verzeichnis: sys:prefs/presets/GoldED, Endung *.api). Nach Bestätigung mit OK steht dann nach (endlich) langer Wartezeit die gewünschten Docks zur Verfügung.

Die *.api-files mit Small im Namen sind zur Verwendung auf einem Hires-Schirm gedacht. Fehlt dieser Zusatz so ist die Verwendung einer Grafikkarte oder eines A2024 empfohlen.

Zur Installation von Hand sind einfach alle Dateien im Verzeichnis GoldDock/ nach GoldED:api/GoldDock zu kopieren. Die Dateien im Verzeichnis ARexx/ gehören nach GoldED:ARexx/. Ferner gibt es im Verzeichnis Presets/ einige Voreinstellungen für API, Menüs, Tastatur, die nach SYS:prefs/presets/GoldED/ kopiert werden können.

Zur Tastatur-/Menubelegung existiert keine Dokumentation. Man kann diese jedoch mit GEDRefs ausgeben lassen. (s. GEDRefs-Verzeichnis in GoldED:Tools/.

Um eines der mitgelieferten Docks zu benutzen, trägt man im API-Requester des GoldED (Menü Konfig/API) im ListView-Gadget als Client GoldED:API/GoldDock/GoldDock sowie die zu verwendende Prefs-Datei ein, z.B

```
GoldED:API/dock/GoldDock PROGDIR:WordDock.prefs
```

(Wird die Prefs-Datei nicht explizit angegeben, so wird nach dock.prefs in PROGDIR: gesucht.)

Alternativ kann auch eines der mitgelieferten API-Konfigurationsfiles geladen werden.

Nach Verlassen des Requesters wird nach den Angaben in der Prefs-Datei ein Dock aufgebaut. Im obigen Beispiel erscheint eine Iconleiste, die die häufigsten Funktionen im Umgang mit Dateien, dem Clipboard und den Fenstern zur Verfügung stellt.

Mittels GoldED können die Einträge in den Prefs-Dateien editiert werden, so daß jeder die Docks nach seinen Wünschen konfigurieren kann.

1.6 API

API (Application Interface)

Mit diesem GoldED-Requester im Konfig-Menü können Sie festlegen, welche GoldED-Clients während des Programmstarts installiert werden sollen. Dies sind externe Programme, die mit dem Editor über ein spezielles Protokoll Informationen austauschen können und so regelmäßig über bestimmte Ereignisse informiert werden.

Mit Clients kann der Funktionsumfang des Editors flexibel erweitert werden, wie man hoffentlich auch am GoldDock feststellt.

Zusammen mit GoldED werden drei Clients samt Beispielcode in C

ausgeliefert:

Dock	eine einfachere, nicht so flexible Implementierung von Docks wie hier bei GoldDock,
Rexx	Beispiel-Client zur Erweiterung des Befehlssatz des Editors
Spell	ein Online-Spellchecker

Weitere Informationen hierzu finden sich in der GoldED-Distribution.

1.7 Dock-Konfiguration

DOCK-KONFIGURATION

Die Dock-Konfiguration ist (zur Zeit) nur über eine ASCII-Konfigurationsdatei möglich. Dies paßt zwar nicht ganz in das Konzept des GoldED, wo fast alle Einstellungen über komfortabel zu bedienende Requester vorgenommen werden können, aber dies kann sich ja bei genügender Resonanz evtl. noch ändern. Alternativ könnte man auch mit kleinen AREXX-Skripten das ganze vereinfachen; was mir allerdings vorschwebt ist eine Konfiguration in der Art wie beim ToolManager.

Erfahrene Anwender können Aussehen, Lage, Orientierung und Funktion der Funktionsleisten jederzeit an die eigenen Wünsche anpassen, indem sie die Dock-Konfigurations-Dateien wie z.B GoldED:API/GoldDock/WordDock.prefs mit GoldED laden und edi(ti)eren.

Hier ein Beispiel für eine Dock-Konfiguration:

```
; Window-Dock-Konfiguration
; und andere Kommentare zum Dock .....

; Definition der Programme
PROG "WindowFreeze" AREXX COMMAND="'ADDRESS %s; FREEZE CURRENT'"
PROG "WindowExchange" AREXX COMMAND="'ADDRESS %s; FREEZE SWAP'"

; Definition der Piktogramme
ICON "Freeze" "Freeze.iff"
ICON "Exchange" "GoldED:api/GoldDock/icons/Exchange.iff"

; Definition der Dock-Einträge
; ein Dock-Eintrag besteht notwendigerweise PROG-Objekt
; sowie evtl. ICON- und SOUND-Objekt
DTool PROG "WindowFreeze" ICON "Freeze"
DTool PROG "WindowExchange" ICON "Exchange"

; Definition des Docks
;
; alle bis zum Auftreten des Befehl DOCK definierten
; Dock-Einträge werden dort eingefügt
DOCK "WindowDock" X=0 Y=0 HORIZONTAL COLUMNS=1 POPUP HOTKEY "rcommand w"
```

In dieser Datei können neben Leerzeilen und Kommentaren (eingeleitet durch ein Semikolon) die Befehle ICON, PROG, SOUND, DTool und DOCK verwendet werden.

Jede Konfigurations-Datei enthält die Definition für EIN Dock, wobei maximal 50 Dock-Einträge zulässig sind. Für mehrere Docks müssen entsprechend mehrere Clients eingetragen werden (s. Tips).

Mit ICON, PROG und SOUND werden die entsprechenden ToolManager-Objekte definiert. Jedes dieser Objekte muß einen eindeutigen Namen zugewiesen bekommen. Dabei wird zwischen Groß- und Kleinschreibung unterschieden.

Sollen diese Objekte in einem Dock verwandt werden, so können sie mit dem DTOOL-Befehl zu einem Dock-Eintrag zusammengefaßt werden. Die Reihenfolge der DTOOL-Befehle legt dabei die Reihenfolge im Dock fest.

Mit DOCK schließlich wird ein Dock-Objekt erzeugt, welches alle zuvor definierten Dock-Einträge enthält. Ferner wird der Name des Objekt sowie dessen Eigenschaften festgelegt (Orientierung, Position, Anzahl Spalten, Art des Dock wie POPUP, CENTER, etc.).

Speichern Sie die Einstellungsdatei nach erfolgter Änderung ab und rufen Sie dann den API-Requester erneut auf: mit Betätigung des OK-Gadgets wird das Dock neu gestartet und damit die neuen Einstellungen übernommen.

1.8 PROG

PROG

Mit PROG werden WB-/CLI-Kommandos, ARexx-Kommandos und -Makros, sowie HOTKEYs definiert. Zwingend anzugeben sind ein Name unter dem das Objekt definiert wird, sowie ein zugehöriger COMMAND-String und der Typ (AREXX, CLI, HOTKEY, WB).

Kommando	Option	Beschreibung
PROG	NAME/A	setzt den Namen des PROG-Objekts
	AREXX/S	setzt Programm-Typ auf AREXX
	CLI/S	- " - CLI
	WB/S	- " - WB/S
	HOTKEY/S	setzt Aktions-Typ auf HOTKEY
	COMMAND/K/A	Kommando (%s wird durch Hostname ersetzt)
	DIR/K	'aktuelles Verzeichnis' während der Aktion
	OUTPUT/K	Ausgabedatei
	KEY/K	Tasten-Beschreibung, über den das Objekt ebenfalls ausgeführt wird (s. Tastenbefehle)
	ARGS/S	sollen Argumente an das PROG übergeben werden, funktioniert nur bei CLI-Kommandos

BEMERKUNGEN:

Allgemeines

Namen, Kommandos und alle übrigen zuzuweisenden Zeichenketten, die

Leerzeichen enthalten sind in Anführungszeichen ("...") zu setzen.

Sollen an GoldED zu schickende Befehle wiederum Anführungszeichen enthalten, so sind diese durch "*" anzugeben. (s. auch die beiliegenden Konfigurationsdateien)

Bei Objektnamen wird zwischen Groß- und Kleinschreibung unterschieden

Typ AREXX

Kommandos in Hochkommata werden vom ARexx-Server als Befehle interpretiert, Kommandos in Anführungszeichen gelten als Makro-Aufruf.

Typ HOTKEY bzw. Argument KEY

Im COMMAND-String bzw. KEY-String muß ein gültiger Input Description String stehen. (Tastenbeschreibung)

1.9 ICON

ICON

Mit ICON werden Bild-Objekte definiert. Die Definition setzt sich lediglich zusammen aus dem Objektnamen sowie einem Dateinamen, woraus die Bilddaten gelesen werden.

Kommando	Option	Beschreibung

ICON	NAME/A	setzt den Namen des Bild-Objekts
	FILE/A	Datei, welche die Bilddaten enthält

Beispiel:

ICON "BildName" "GoldED:API/dock/icons/Bild.iff"

BEMERKUNG:

An Bilddaten werden IFF-Bilder, -Animationen sowie .info-Dateien unterstützt.

1.10 SOUND

SOUND (noch nicht getestet, muesste aber funktionieren)

Mit SOUND werden Ton-Objekte definiert. Die Definition setzt sich zusammen aus dem Objektnamen, einem Befehl, sowie dem ARexx-Port, an den der Befehl gesandt werden soll. Dies ist nötig, da kein eigener Geräuschabspieler im Programm integriert ist.

Stattdessen kann jedes externe Abspielprogramm benutzt werden, welches über

einen ARexx-Port verfügt, z.B. upd, welches auch in der ToolManager-Distribution enthalten ist.

Kommando	Option	Beschreibung
SOUND	NAME/A	setzt den Namen des Ton-Objekts
	COMMAND/K/A	Befehl, welcher an den ARexx-Port geschickt werden soll
	PORT/K	Name des ARexx-Port des externen Abspielprgs; Default: PLAY (Port von upd)

1.11 DTOOL

DTOOL

Mit DTOOL werden Objekte vom Typ PROG sowie optional ICON und SOUND zur Verwendung in einem Dock zusammengefaßt. Die Reihenfolge in der Konfigurationsdatei legt dabei diejenige im Dock fest.

DTOOL

Kommando	Option	Beschreibung
DTOOL	PROG/K/A	Name eines Programm-Objekts
	ICON/K	Name eines Bild-Objekts
	SOUND/K	Name eines Ton-Objekts

BEMERKUNG:

Dock-Einträge, die nur aus einem Programm-Objekt bestehen, sind sinnvoll einzusetzen in Text-Docks (Option TEXT beim DOCK-Befehl). Statt eines Bildes erscheint dann als Dock-Eintrag der Name des Prog-Objekts. Benutzt man hingegen kein Text-Dock, so bleibt der zugehörige Bereich im Dock leer. Gegebenfalls erscheint dort ein Muster (Option PATTERN beim DOCK-Befehl).

1.12 DOCK

DOCK

Der DOCK-Befehl definiert ein Dock, welches alle zuvor in der Konfigurations-Datei aufgetretenen Dock-Einträge (s. DTOOL-Befehl) enthält. Ferner werden die Eigenschaften des Docks festgelegt. So können z.B. Text-Docks erstellt werden. Die einzelnen Gadgets sind dann mit den Namen der Programm-Objekte beschriftet. Ferner kann entschieden werden, ob das DOCK direkt sichtbar ist oder nicht, wo es sich öffnet, die Orientierung, Anzahl der Spalten, Titelzeile, ob es sich automatisch nach Auswahl eines Gadgets wieder schließt, u.v.a.m. ...

Kommando	Option	Beschreibung

DOCK	NAME/A	Name des Dock-Objekts
	X=XPOS/N	x-Position des Docks (Def.: rechter Rand)
	Y=YPOS/N	y-Position des Docks (Def.: unterhalb des Schirmtitel)
	V=VERTICAL/S	Richtung der Dragbar des Docks (Def.: Horizontal)
	COLS=COLUMNS/N	Anzahl der Spalten
	TEXT/S	Text-Dock mit Zeichensatz des Schirms
	CENTER/S	Dock erscheint unter Mauszeiger
	POPUP/S	Dock verschwindet nach Auswahl
	HOTKEY/S	Tastenbefehl zum Togglen des Zustands des Dock (sichtbar/nicht sichtbar); zur Zeit nötig, damit sich Docks gegenseitig aufrufen, da noch nicht mehrere Dock-Objekte in einer Konfigurationsdatei möglich sind.
	HIDDEN/S	Dock nicht sichtbar; kann über HOTKEY aktiviert werden
	FM=FRONTMOST/S	öffnet Dock auf dem vordersten öffentlichen Schirm; ACHTUNG: Nicht setzen, falls automatisch der Schirm ausgewählt werden soll, auf dem der GoldED läuft
	TITLE/K	Name des Dock-Fensters; dadurch werden automatisch Dragbar, Close- und Depth-Gadget sowie ein Rand hinzugefügt
	PATTERN/S	Einträge, die kleiner als der größte Dock-Eintrag sind, werden mit einem Muster hinterlegt.
	STICKY/S	Das Dock merkt sich die Position, an der es geöffnet wurde, kann zwar verschoben werden, öffnet sich aber immer wieder an der ursprünglichen Stelle

BEMERKUNGEN:

X und Y werden bei Angabe von CENTER ignoriert.

1.13 Tips

TIPS

1) Verwendung mehrerer Docks

Es können mehrere Docks erzeugt werden, indem für jedes Dock ein eigener Eintrag im API-Requester eingesetzt wird. So kann man z.B. Docks erzeugen, die sich über einen Hotkey gegenseitig aufrufen, d. h. der Zustand des Docks wird jedesmal geändert (sichtbar/nicht sichtbar)

Um unnötigen Speicherverbrauch zu vermeiden, sollte man bei mehreren Docks GoldDock resident machen. Dann muß allerdings bei der Prefs-Datei der Pfad angegeben werden, der Client lautet nun nur GoldDock. Der Eintrag im APIRequester könnte also lauten:

```
GoldDock GoldED:API/dock/WordDock.prefs
```

2) Reaktivierung des GoldED-Fensters

Bei Betätigung eines Dock-Piktogramms ist zunächst das Dock-Window aktiv. Um wieder das zuletzt aktive GoldED-Fenster zu reaktivieren, ist in ein "WINDOW FORCE" an den GoldED zu schicken. Ein AREXX-PROG-Objekt könnte dann lauten:

```
PROG "ClipCut" AREXX COMMAND="'ADDRESS %s; WINDOW FORCE; CLIP CUT"
```

1.14 AREXX-Skripte

WindowArrange.ged

Das AREXX-Skript ordnet die GoldED-Fenster an unter Beachtung von am Anfang des Skripts definierten freizulassenden Rändern auf dem Schirm. Als Argument kann man die Priorität für das aktive Fenster angeben.

Man kann so z.B. in die oberen Fensterränder Gadgets setzen lassen, die die Fenster arrangieren mit der angegebenen Priorität für das Fenster, in welchem dieses Gadget betätigt wurde. Dies ist im mitgelieferten WindowArrange.men (Verzeichnis Presets) demonstriert, welches an die eigenen Menus angehängt werden kann.

AGReference.ged

Dieses Skript stellt eine Basis für ein auf AmigaGuide basierendes Referenzsystem dar. Es kann gut auf eine Taste wie Shift Help gelegt werden. Nach Aufruf des Skript erscheint dann das Wort unter dem Cursor bzw. der Funktionsname in einem StringRequester (benötigt die REXXReq.library). Über die verschiedenen Buttons kann dann ausgewählt werden in welchen Referenzfile das Wort gesucht werden soll. Die Art, wie das Wort ausgewählt wird, hängt unter anderem davon ab, welche Leerzeichen im GUI-Requester von GoldED eingetragen sind.

Die verwendeten Referenzfiles müssen im Suchpfad von AmigaGuide liegen oder im Skript angepaßt werden. Für Autodocs ist voreingestellt s:autodocs.xref, für AREXX ARx_Guide.xref, für Dice dcc:man/dcc.xref. GoldED sucht den entsprechenden Node in der GoldED-Anleitung und Hyperdocs: versucht eine Datei mit dem Namen des Wortes zzgl. der Extension .guide im Suchpfad von AmigaGuide zu öffnen.

ARx_Guide.xref liegt dem Paket AREXXGuide20.lha (AmiNet) bei. Es stellt eine Dokumentation und Referenz zur Programmierung in AREXX zur Verfügung.

ARx_Help.ged

Dies ist eine für GoldED angepaßte Version der bei AREXXGuide 2.0 ((C) 1993/1994 Robin Evans) mitgelieferten Editorsripten für eine Online-Hilfe. Diese ist sehr nützlich zur Programmierung in (A)REXX.

HiSpeed.ged

Wird benötigt, um vom WordDock aus HiSpeed zu starten, wobei der Name des im aktuellen Fensters befindlichen Textes übergeben wird.

InsertVar.ged

Fügt die im Argument übergebene, im GoldED mit QUERY zu erfragende Variable, an der aktuellen Stelle im Text ein. Wird z.B. von WordDock für das Einfügen der aktuellen Zeit bzw. Datums verwandt.

Verzeichnis TeXSkripte in ARexx

Hier sind eine Reihe Makros für die Zusammenarbeit mit PasTeX vorzufinden. Sie sind erstellt und evtl. von mir leicht modifiziert von René Laederach, Fin Schuppenhauer, Georg Hessmann, Jörg Höhle u.a. (s. Makros) Sie werden vom TeXDock bzw. von PasTeX.men verwandt.

1.15 PRESETS

PRESETS

In das Verzeichnis sys:prefs/presets/GoldED werden durch das Installer-Skript einige meiner GoldED-Konfigurationen abgelegt. Ihre Namen beginnen alle mit ArmaKonfig. Die zugehörige Belegung ist mit GEDRefs aus dem Verzeichnis GoldED:Tools/GEDRefs automatisch ausgegeben worden.

Die Belegung richtet sich nach Möglichkeit nach den mir bekannten Vorschlägen des StyleGuides bzw. der Eindeutschung der Workbench durch Commodore.

Tastaturbelegung
Menubelegung

1.16 KEYLAYOUT

Die Shortcuts für die mitgelieferten Docks sind wie folgt:

WordDock	kein shortcut :-)
BlockDock	ctrl right mouse button
MacroDock	ctrl shift m
PrefsDock	ctrl shift p
GUIMakeDock	ctrl shift d
TeXDock	ctrl shift t
SumDock	ctrl +
BracketDock	ctrl (
GreekLetterDock	ctrl shift g

Die beiliegende Konfiguration ArmaKonfig.key liefert folgende Tastaturbelegung:

<A> <S> BACKTAB	INTERNAL: TAB BACK FIXED
<A> <S> DEL	INTERNAL: INSERT WORD
<A> <S> LEFT	INTERNAL: PREVEND
<A> <S> RIGHT	INTERNAL: ENDWORD

<A> DEL	INTERNAL: DELETE WORD
<A> DOWN	INTERNAL: FDOWN
<A> LEFT	INTERNAL: PREV
<A> RIGHT	INTERNAL: NEXT
<A> TAB	INTERNAL: TAB FIXED
<A> UP	INTERNAL: FUP
<C> 010	INTERNAL: INSERT LINE
<C> 014	INTERNAL: FIND NEXT
<C> 016	INTERNAL: FIND PREV
<C> 022	INTERNAL: CLIP VPASTE
<C> <A> <S> RIGHT	INTERNAL: GOTO EOL
<C> <S> LEFT	INTERNAL: SHIFT LEFT TAB
<C> <S> RIGHT	INTERNAL: SHIFT RIGHT TAB
<C> CR	INTERNAL: INSERT LINE
<C> DEL	INTERNAL: POP
<C> DOWN	INTERNAL: GOTO BOTTOM
<C> F01	REXX: GoldED:Arexx/InsertICON GoldED:API/GoldDock/ ↵
Icons/	
<C> HELP	INTERNAL: FOLD TOGGLE ALL
<C> LEFT	INTERNAL: FIRST
<C> RIGHT	INTERNAL: GOTO EOL
<C> UP	INTERNAL: GOTO TOP
<N> .	INTERNAL: DEL
<N> 0	INTERNAL: MODE INSERT TOGGLE
<N> 1	INTERNAL: GOTO BOTTOM
<N> 2	INTERNAL: DOWN
<N> 3	INTERNAL: DPAGE
<N> 4	INTERNAL: LEFT
<N> 6	INTERNAL: RIGHT
<N> 7	INTERNAL: GOTO TOP COLUMN 1
<N> 8	INTERNAL: UP
<N> 9	INTERNAL: UPAGE
<N> CR	INTERNAL: CR
<S> BACK	INTERNAL: DELETE LEFT
<S> BACKTAB	INTERNAL: TAB BACK
<S> CR	INTERNAL: SMARTCR
<S> DEL	INTERNAL: DELETE EOL
<S> DOWN	INTERNAL: DJUMP
<S> ESC	INTERNAL: CMD
<S> F01	INTERNAL: SAVE ALL ASK
<S> F02	INTERNAL: SAVE ALL ASK CRUNCH
<S> F03	INTERNAL: QUIT
<S> F04	INTERNAL: NEW
<S> F05	INTERNAL: MARK SET
<S> F06	INTERNAL: REPLACE ASK
<S> F07	INTERNAL: UPAGE
<S> F08	INTERNAL: MODE INSERT FALSE
<S> F09	INTERNAL: PROJECT ASK
<S> F10	INTERNAL: MACRO RECORD
<S> HELP	REXX: golded:arexx/AGReference output: nil:
	DOS event(s): asynchronous
<S> LEFT	INTERNAL: VLEFT
<S> RIGHT	INTERNAL: VRIGHT
<S> TAB	INTERNAL: TAB BACK
<S> UP	INTERNAL: UJUMP
BACK	INTERNAL: BACK

CR	INTERNAL: CR
DEL	INTERNAL: DEL
DOWN	INTERNAL: DOWN
ESC	INTERNAL: PHRASE CURRENT
F01	INTERNAL: OPEN FAST ASK OLDPATH
F02	INTERNAL: OPEN INSERT ASK
F03	INTERNAL: PRINT ALL
F04	INTERNAL: MARK HIDE
F05	INTERNAL: MARK SET
F06	INTERNAL: FIND NEXT
F07	INTERNAL: DPAGE
F08	INTERNAL: MODE INSERT TRUE
F09	INTERNAL: TABS SOLID TOGGLE
F10	INTERNAL: MACRO PLAY
HELP	INTERNAL: FOLD TOGGLE
LEFT	INTERNAL: LEFT
RIGHT	INTERNAL: RIGHT
TAB	INTERNAL: TAB
UP	INTERNAL: UP

1.17 MENULAYOUT

BEMERKUNG !!!

Die Menus duerften auf einem HIRES-Schirm zu umfangreich sein. Daher sollten nicht benötigte Menus einfach im entsprechenden Requester entfernt werden.

MENUS:

[Projekt]

Neues Fenster	INTERNAL: MORE SMART
öffnen schnell ...	shortcut: R-Amiga N INTERNAL: OPEN FAST ASK OLDPATH
öffnen neu ...	shortcut: R-Amiga O INTERNAL: OPEN NEW ASK
öffnen ...	INTERNAL: OPEN ASK OLDPATH
öffnen Original	INTERNAL: OPEN AGAIN
speichern	shortcut: R-Amiga * INTERNAL: SAVE ALL
speichern als ...	shortcut: R-Amiga S INTERNAL: SAVE ALL ASK
speichern XPK als ...	shortcut: R-Amiga A INTERNAL: SAVE ALL ASK CRUNCH
speichern & beenden	INTERNAL: SAVE ALL EXIT
Schutzbits ...	shortcut: R-Amiga D INTERNAL: BITS ASK
Text löschen	INTERNAL: NEW NONAME
Einfügen ...	shortcut: R-Amiga Z INTERNAL: OPEN INSERT ASK
Anhängen ...	shortcut: R-Amiga I INTERNAL: OPEN APPEND ASK
Akt. Verzeichnis ...	INTERNAL: DIR ASK
Neuer Name ...	INTERNAL: NAME ASK

Fenster schließen	INTERNAL: QUIT
drucken ...	shortcut: R-Amiga Q INTERNAL: PRINT ALL
über GoldED ...	shortcut: R-Amiga P INTERNAL: INFO VERSION
Benutzer ...	INTERNAL: INFO USER
beenden & entfernen	INTERNAL: QUIT UNLOAD

[Block]

Markieren	INTERNAL: MARK SET shortcut: R-Amiga B
Markierung aus	INTERNAL: MARK HIDE shortcut: R-Amiga H
Ausschneiden	(SUBMENU HEADER)
- Unit 0	INTERNAL: CLIP CUT shortcut: R-Amiga X
- Unit 1	INTERNAL: CLIP CUT UNIT 1
- Unit 2	INTERNAL: CLIP CUT UNIT 2
Kopie anlegen	(SUBMENU HEADER)
- Unit 0	INTERNAL: CLIP COPY shortcut: R-Amiga Y
- Unit 1	INTERNAL: CLIP COPY UNIT 1
- Unit 2	INTERNAL: CLIP COPY UNIT 2
Einfügen	(SUBMENU HEADER)
- Unit 0	INTERNAL: CLIP PASTE shortcut: R-Amiga V
- Unit 1	INTERNAL: CLIP PASTE UNIT 1
- Unit 2	INTERNAL: CLIP PASTE UNIT 2
Einfügen vertikal	(SUBMENU HEADER)
- Unit 0	INTERNAL: CLIP VPASTE shortcut: R-Amiga <C> V DOS event(s): asynchronous shanghai time: 0
- Unit 1	INTERNAL: CLIP VPASTE UNIT=1
- Unit 2	INTERNAL: CLIP VPASTE UNIT=2
Kopieren	INTERNAL: BLOCK COPY shortcut: R-Amiga C
Löschen	INTERNAL: DELETE BLOCK shortcut: R-Amiga \
Verschieben	INTERNAL: BLOCK MOVE shortcut: R-Amiga M
Spalte löschen	INTERNAL: DELETE COLUMN shortcut: R-Amiga [
Spalte einfügen	INTERNAL: INSERT BLOCK COLUMN shortcut: R-Amiga]
Text anhängen...	INTERNAL: INSERT BLOCK APPEND
Spaltentext...	INTERNAL: INSERT BLOCK shortcut: R-Amiga ?
Einrücken...	INTERNAL: SHIFT ASK
Sortieren	INTERNAL: BLOCK SORT
Gross	INTERNAL: BLOCK UPPER
Klein	INTERNAL: BLOCK LOWER
Sichern als...	INTERNAL: SAVE BLOCK ASK shortcut: R-Amiga K
Drucken...	INTERNAL: PRINT BLOCK

[Layout]

rechten Rand setzen	INTERNAL: LAYOUT CURRENT
Rand übernehmen	INTERNAL: LAYOUT AUTO TOGGLE
	checkmark driven by ALEFT
WordWrap	INTERNAL: LAYOUT WRAP=TOGGLE
	checkmark driven by WRAP
Block bündig	INTERNAL: FORMAT MARK BLOCK
Block links	INTERNAL: FORMAT MARK LEFT
Block rechts	INTERNAL: FORMAT MARK RIGHT
Block zentriert	INTERNAL: FORMAT MARK CENTER
Blocksatz	INTERNAL: FORMAT LINES BLOCK
	shortcut: R-Amiga =
linksbündig	INTERNAL: FORMAT LINES LEFT
	shortcut: R-Amiga U
rechtsbündig	INTERNAL: FORMAT LINES RIGHT
zentriert	INTERNAL:
	shortcut: R-Amiga \$
Autom. Groß/Klein	INTERNAL: MODE AUTOCASE TOGGLE
	checkmark driven by CHKCASE

[Suchen]

Suchen...	INTERNAL: FIND ASK
	shortcut: R-Amiga F
Weitersuchen	INTERNAL: FIND NEXT
	shortcut: R-Amiga <C> N
	DOS event(s): asynchronous
	shanghai time: 0
Vorheriges	INTERNAL: FIND PREV
	shortcut: R-Amiga <C> P
	DOS event(s): asynchronous
	shanghai time: 0
Ersetzen...	INTERNAL: REPLACE ASK
	shortcut: R-Amiga R
Weiter	INTERNAL: REPLACE NEXT
	shortcut: R-Amiga +
	shortcut: R-Amiga CTRL-P
	DOS event(s): asynchronous
	shanghai time: 0
Zählen...	INTERNAL: FIND COUNT ASK
Prüfe ((...))	INTERNAL: BRACKET CHECK
	shortcut: R-Amiga (
passende Klammer	INTERNAL: BRACKET MATCH
	shortcut: R-Amiga)
Referenz	INTERNAL: XREF CURRENT
	shortcut: R-Amiga \$
Referenz ...	INTERNAL: XREF ASK
	shortcut: R-Amiga 7
Ergänzen	INTERNAL: PHRASE CURRENT
ASCII-Tabelle ...	INTERNAL: CODE TABLE
ASCII einfügen ...	INTERNAL: CODE ASK
ASCII-Code zeigen	INTERNAL: CODE SHOW
	shortcut: R-Amiga 2
Zeichensatz ...	INTERNAL: REMAP ASK
Code einfügen	(SUBMENU HEADER)
- ESC (Escape)	TEXT: %27

- FF (Form Feed)	TEXT: %12
Zeichen groß/klein	INTERNAL: CODE TOGGLE
Funktionen ...	shortcut: R-Amiga 0
- alle anzeigen...	(SUBMENU HEADER)
	INTERNAL: FUNC SMART
	shortcut: R-Amiga /
- aktuelles Wort	INTERNAL: FUNC CURRENT SMART
- C-Funktionen...	INTERNAL: FUNC C
- C-Strukturen...	INTERNAL: FUNC STRUCT
- AutoDoc-Einträge	INTERNAL: FUNC AUTODOC
[Kontrolle]	
Gehe zu...	INTERNAL: GOTO ASK
	shortcut: R-Amiga G
Anfang/Ende	INTERNAL: GOTO OTHEREND UNFOLD=FALSE
	shortcut: R-Amiga T
letzte Änderung	INTERNAL: GOTO CHANGE
	shortcut: R-Amiga E
Position merken	(SUBMENU HEADER)
- Position 1	INTERNAL: PING SLOT 1
	shortcut: R-Amiga !
- Position 2	INTERNAL: PING SLOT 2
- Position 3	INTERNAL: PING SLOT 3
- Position 4	INTERNAL: PING SLOT 4
- Position 5	INTERNAL: PING SLOT 5
	shortcut: R-Amiga !
Alte Position	(SUBMENU HEADER)
- Position 1	INTERNAL: PONG SLOT 1
	shortcut: R-Amiga 1
- Position 2	INTERNAL: PONG SLOT 2
- Position 3	INTERNAL: PONG SLOT 3
- Position 4	INTERNAL: PONG SLOT 4
- Position 5	INTERNAL: PONG SLOT 5
	shortcut: R-Amiga 1
Alles falten	INTERNAL: FOLD ALL OPEN=FALSE
Alles entfalten	INTERNAL: FOLD ALL OPEN=TRUE
Einfügen	INTERNAL: MODE INSERT TOGGLE
	checkmark driven by INSERT
	shortcut: R-Amiga ^
TAB's einfügend	INTERNAL: TABS SOLID TOGGLE
	checkmark driven by SOLID
	shortcut: R-Amiga #
NumPad = Bewegung	INTERNAL: MODE NUMPAD TOGGLE
	checkmark driven by USER1
Arrangieren	(SUBMENU HEADER)
- vertikal	INTERNAL: WINDOW ARRANGE 0
- horizontal	INTERNAL: WINDOW ARRANGE 1
Zentrieren	INTERNAL: WINDOW CENTER
Groß/Klein	INTERNAL: WINDOW ZIP
	shortcut: R-Amiga ,
Vergrößern	INTERNAL: WINDOW MAX
	shortcut: R-Amiga -
Übersicht	INTERNAL: VIEW PREVIEW TOGGLE
	checkmark driven by
Einfrieren	(SUBMENU HEADER)
- Einfrieren	INTERNAL: FREEZE CURRENT

```

- Öffnen...          shortcut: R-Amiga <
                     INTERNAL: FREEZE ASK
- Tauschen...        shortcut: R-Amiga >
                     INTERNAL: FREEZE SWAP
Nächstes Fenster     shortcut: R-Amiga _
Iconify              INTERNAL: WINDOW NEXT
                     INTERNAL: SCREEN ICONIFY=TRUE
                     shortcut: R-Amiga :

[Diverses]

Hilfe...            INTERNAL: HELP TOPIC "MAIN"
Sourcefiles...      INTERNAL: PROJECT ASK
Filter...           INTERNAL: GREP ASK
                     shortcut: R-Amiga 3
Datei suchen        INTERNAL: HUNTER CURRENT
                     shortcut: R-Amiga ~
Zeile entfernen     INTERNAL: POP
                     shortcut: R-Amiga 5
Wieder einfügen     INTERNAL: PUSH
                     shortcut: R-Amiga 6
Zeilen tauschen     INTERNAL: LINES SWAP
Zeile verdoppeln    INTERNAL: LINES DOUBLE
                     INTERNAL: DOWN
Alte Zeile          INTERNAL: UNDO
                     shortcut: R-Amiga |
Statistik           INTERNAL: INFO TEXT
Neue Shell          DOS: "newshell con:" \CON "/Shell/AUTO/WAIT/CLOSE/ ←
                     SCREEN" \SCREEN " FROM S:SHELL-STARTUP"
                     shortcut: R-Amiga `
                     DOS event(s): asynchronous
Dateien ...         (SUBMENU HEADER)
- löschen...        INTERNAL: FILE DELETE
- umbenennen....    INTERNAL: FILE RENAME
- Verzeichnis anlegen... INTERNAL: FILE NEWDIR
Rechner             DOS: "SYS:TOOLS/CALCULATOR"
                     shortcut: R-Amiga %
                     DOS event(s): asynchronous
                     shanghai time: 5
HiSpeed ...         DOS: "HISPEED:HISPEED " %34 \NAME %34 " NOPRINT ←
                     SCREEN " \SCREEN

                     DOS event(s): asynchronous
Datum einfügen      TEXT: $USERTOWN ", den " \DATE
Zeit einfügen       TEXT: "Zeit: " \TIME
Pfad einfügen ...   INTERNAL: INSERT PATH
Kommando ...        INTERNAL: CMD
Letzter Fehler ...  INTERNAL: INFO ERROR

[Makros]

Makro edieren...    INTERNAL: OPEN ASK PATH="GoldED:ARexx"
als Makro starten    REXX: \NAME
Aufnahme Start/Ende INTERNAL: MACRO RECORD
Abspielen           INTERNAL: MACRO PLAY
öfter abspielen...  INTERNAL: MACRO PLAY ASK
Öffnen...           INTERNAL: MACRO LOAD
Speichern als ...    INTERNAL: MACRO SAVE

```

```

dmake                                REXX: GoldED:ARexx/dmake
output: "con:" \CON " \DMake/AUTO/SCREEN" \SCREEN
Übersetzen (DCC)                    REXX: GoldED:ARexx/dcc
Makros                              (SUBMENU HEADER)
- markiere {...}                    REXX: GOLDED:ARexx/block
- Block falten                      REXX: GOLDED:ARexx/fold
- C-Funktion...                     REXX: GOLDED:ARexx/func TRUE
                                   shortcut: R-Amiga 4
- switch...                         REXX: GOLDED:ARexx/switch
- case...                          REXX: GOLDED:ARexx/case
- Revision...                      REXX: GOLDED:ARexx/rev FALSE FALSE TRUE
- Numerieren...                   REXX: GOLDED:ARexx/number
- Assemblieren                    REXX: GOLDED:ARexx/a68k
- Zeile ausführen                 REXX: GOLDED:ARexx/perform
- Pfad übernehmen                 REXX: GOLDED:ARexx/getpath
----- GUIMake -----
Konfig...                           INTERNAL: REQUEST BODY "@'93 Rico Krasowski"
    AREXX/GUI                       INTERNAL: RX PORT=GUIMAKE CMD=CONFIG MACRO=GOLDED: ↵
Module...                           INTERNAL: RX PORT=GUIMAKE CMD=MODULES MACRO=GOLDED: ↵
    AREXX/GUI
Optionen...                         INTERNAL: RX PORT=GUIMAKE CMD=OPTIONS MACRO=GOLDED: ↵
    AREXX/GUI
Übersetzen/linken                  INTERNAL: RX PORT=GUIMAKE CMD=BUILD MACRO=GOLDED: ↵
    AREXX/GUI
neu übersetzen                     INTERNAL: RX PORT=GUIMAKE CMD="BUILD ALL" MACRO= ↵
    GOLDED:AREXX/GUI
Fehler                             (SUBMENU HEADER)
- alle anzeigen...                 REXX: GOLDED:ARexx/errors
- Fehler: erster                   INTERNAL: RX PORT=GUIMAKE CMD=FIRST MACRO=GOLDED: ↵
    AREXX/GUI
- Fehler: nächster                INTERNAL: RX PORT=GUIMAKE CMD=NEXT MACRO=GOLDED: ↵
    AREXX/GUI
- Fehler: vorheriger              INTERNAL: RX PORT=GUIMAKE CMD=PREV MACRO=GOLDED: ↵
    AREXX/GUI
Executable starten                 INTERNAL: RX PORT=GUIMAKE CMD=RUN MACRO=GOLDED: ↵
    AREXX/GUI

[Konfig]

Tastatur...                        INTERNAL: BIND
Tab's...                           INTERNAL: TABS ASK
Referenzen...                      INTERNAL: PATH
Pfade...                          INTERNAL: HUNTER ASK
Wörterbuch...                     (SUBMENU HEADER)
- auswählen...                    INTERNAL: PHRASE ASK
    AREXX                          INTERNAL: PHRASE CONFIG sys:prefs/presets/Golded/ ↵
    AREXX.dic
- C                                INTERNAL: PHRASE CONFIG sys:prefs/presets/Golded/C. ↵
    dic
- E                                INTERNAL: PHRASE CONFIG sys:prefs/presets/Golded/E. ↵
    dic
- LaTeX                           INTERNAL: PHRASE CONFIG sys:prefs/presets/Golded/ ↵
    LaTeX.dic
Einrückungen..                    INTERNAL: INDENT ASK
Anzeige...                        INTERNAL: SCREEN ASK
Menus...                          INTERNAL: MENUS ASK
Oberfläche...                     INTERNAL: GUI ASK

```

```

API... INTERNAL: API ASK
output: CON:///Debug/AUTO
Layout... INTERNAL: LAYOUT ASK
shortcut: R-Amiga &
Drucker... INTERNAL: PRINT ASK
Diverses... INTERNAL: MISC ASK
Speichern... INTERNAL: PREFS SAVE
Laden... INTERNAL: PREFS LOAD
Speichern INTERNAL: PREFS SAVE FILE envarc:GoldED/GoldED. ↵
  prefs
Konfiguration splitten INTERNAL: PREFS SPLIT sys:prefs/presets/MyGoldEd/ ↵
  MyKonfig

[PasTeX]

TeX-Server ... (SUBMENU HEADER)
- starten REXX: GoldED:ARexx/TeXSkripte/TeX-Server.ged \ ↵
  SCREEN
output: "CON:100/725/800/299/PasTeX-Server-Shell/ ↵
  AUTO/SCREEN/" \SCREEN
- schließen REXX: GoldED:ARexx/TeXSkripte/Quit_TeX.ged
Konfiguration (SUBMENU HEADER)
- Format ... DOS: tex:bin/texformat currentformat=texformat ↵
  formatpath=tex:formats pattern=#?.fmt
output: nil:
DOS event(s): asynchronous
- SmallTeX DOS: rxset virtex tex:bin/virtex
output: nil:
DOS event(s): asynchronous
- SmallTeX resident DOS: GoldED:ARexx/TeXSkripte/SmallTeXResident
output: NIL:
DOS event(s): asynchronous
- BigTeX DOS: rxset tex:bin/BigVirTeX
DOS event(s): asynchronous
- BigTeX resident DOS: Golded:ARexx/TeXSkripte/BigTeXResident
output: NIL:
- Fonts direkt erzeugen REXX: GoldED:ARexx/TeXSkripte/SetCALLMF.ged
checkmark driven by USER1
Dokument übersetzen REXX: golded:arexx/TeXSkripte/Start_TeX.ged
DOS event(s): asynchronous
SpecialHost starten DOS: tex:bin/SpecialHost
DOS event(s): asynchronous
ShowDVI in Vordergrund REXX: GoldED:ARexx/TeXSkripte/SDVIToFront.rexx
DVIPrint-Setup REXX: tex:rexx/StartDVIPrintSetUp.ged
DOS event(s): asynchronous
Makros (SUBMENU HEADER)
- TeX-Umlaute einsetzen REXX: GoldEd:ARexx/TeXSkripte/TeXUmlaute.ged
- Goto SDVI REXX: GoldEd:ARexx/TeXSkripte/GoToSDVI.ged
- NextError REXX: Golded:ARexx/texmacrosrene/NextError.ged
- DVI-File drucken... REXX: golded:arexx/PasTeX/Startdviprint.ged
Dokumentation ... (SUBMENU HEADER)
- PasTeX DOS: ShowDVI tex.dvi
output: nil:
directory: tex:doc
DOS event(s): asynchronous
- ShowDVI DOS: ShowDVI sdvi.dvi
output: nil:

```

```

        directory: tex:doc
        DOS event(s): asynchronous
- ShowDVI-Guide
        DOS: amigaguide showdvi.guide
        output: nil:
        directory: tex:config
        DOS event(s): asynchronous
- DVIPrint
        DOS: ShowDVI dvip.dvi
        output: nil:
        directory: tex:doc
        DOS event(s): asynchronous

[ARexx]

ARexx-Referenz
        REXX: GoldED:ARexx/ARx_Help.ged
        DOS event(s): asynchronous
ARexx-Wörterbuch
        INTERNAL: ?
Trace-Konsole ein
        DOS: tco
        DOS event(s): asynchronous
Trace-Konsole aus
        DOS: tcc
        DOS event(s): asynchronous

[Fenster]

Arrangieren
        (SUBMENU HEADER)
- 3x
        REXX: golded:arexx/WindowArrange 3 100 105 28 0 23
        (leave out)
- 2x
        REXX: golded:arexx/WindowArrange 2 100 105 28 0 23
        (leave out)
        shortcut: R-Amiga .
- 1x
        REXX: golded:arexx/WindowArrange 1 100 105 28 0 23
        (leave out)
- vertical
        INTERNAL: WINDOW ARRANGE 0
- horizontal
        INTERNAL: WINDOW ARRANGE 1
Zwischenablage ...
        INTERNAL: OPEN NEW FORCE NAME "Unit 0" QUIET
        INTERNAL: WINDOW CENTER
        INTERNAL: FILE NAME "Inhalt der Zwischenablage 0"
        INTERNAL: CLIP PASTE

[ ? ]

GoldEd ...
        (SUBMENU HEADER)
- Befehle...
        DOS: AmigaGuide Anleitung Document BEFEHLSLISTE
        directory: GoldED:
        DOS event(s): asynchronous
- History
        DOS: AmigaGuide golded:History
        output: NIL:
        DOS event(s): asynchronous
        output: NIL:
        DOS event(s): asynchronous
SpellIt
        DOS: Amigaguide golded:api/spellit/anleitung
        DOS event(s): asynchronous
HiSpeed ...
        (SUBMENU HEADER)
- Anleitung
        DOS: AmigaGuide HiSpeed:Anleitung
        DOS event(s): asynchronous
- ARexxDoc
        DOS: AmigaGuide HiSpeed:ARexxDoc
        output: NIL:
        DOS event(s): asynchronous

```

```

- Setup                                DOS: AmigaGuide HiSpeed:Setup
                                        output: NIL:
                                        DOS event(s): asynchronous
                                        output: NIL:
                                        DOS event(s): asynchronous
AutoDocs ...                           (SUBMENU HEADER)
- DOS                                  DOS: amigaguide dos
                                        output: nil:
                                        DOS event(s): asynchronous
- ToolManager                          DOS: AmigaGuide ToolManager
                                        output: nil:
                                        DOS event(s): asynchronous
- ReqTools                             DOS: amigaguide reqtools
                                        output: nil:
                                        DOS event(s): asynchronous
                                        output: nil:
                                        DOS event(s): asynchronous
Andere Programme ...                   DOS: AmigaGuide "dir.guide"
                                        output: nil:
                                        DOS event(s): asynchronous
Neue Guide-Datei                       INTERNAL: OPEN guides:dir.guide SMART
AG-Referenz...                         REXX: golded:arexx/AGReference.ged
                                        DOS event(s): asynchronous

```

1.18 PLÄNE

PLÄNE

Bei genügend großer Resonanz werde ich mich darum bemühen, daß die Dock-Konfiguration vereinfacht wird. Dies kann zum einen bedeuten, daß die Beschränkungen bei den Prefs-Files entfallen, sowie evtl. ein Preferences-Requester, der aus dem GoldED heraus aufgerufen werden kann.

Für Anregungen oder die Bereitschaft dieses Programm weiterzuentwickeln wäre ich sehr dankbar, da bei mir absehbar ist, daß ich in nächster Zeit aufgrund von Prüfungsvorbereitungen nicht die allermeiste Zeit haben werde.

1.19 DANKSAGUNG

DANKSAGUNG

Zunächst einmal möchte ich mich an dieser Stelle noch einmal ganz herzlich bei Dietmar Eilert für den GoldED bedanken, sowie seine Bereitschaft bei allen auftretenden Problemen/Wünschen unkompliziert und schnell zu helfen.

Ein weiterer Dank geht an St. Becker für den ToolManager, den ich nicht mehr missen möchte und der die Entwicklung des GoldDock sehr einfach machte.

Ferner gilt ein besonderer Dank Suzanne Hoffmann, deren Modem mir wertvolle Dienste geleistet hat und der ich für die (nahe) Zukunft alles Gute mit ihrem Amerikaner (:)) wünsche, sowie meiner Freundin Susanne für ihre Geduld.

Und zu guter letzt noch eine Träne für Commodore International, die uns den Amiga in seiner jetzigen Form brachten.

RIP :- (

1.20 ADRESSE

ADRESSE

Bugreports, Kommentare oder Vorschläge werden jederzeit gerne entgegengenommen.

Meine Adresse lautet

Markus Aretz
Bismarckstr. 20
50672 Köln

Telefon: 0221 / 512503
Fax: 02181 / 73467

EMail: ma@ph-cip.uni-koeln.de (InterNet).

Bankverbindung:

Kto Nr. 2319614 00
BLZ 370 700 60 (Dt. Bank Köln)

1.21 Tastenbefehle

Hinweis:

Die folgenden Beschreibung der Tastenbefehle ist der Anleitung zum ToolManager 2.1 entnommen.

Wie man einen Tastenbefehl definiert

Dieses Kapitel beschreibt wie man einen Tastenbefehl als einen Input Description String definiert, der dann von Commodities ausgewertet werden kann. Jedes Mal, wenn ein Tastenbefehl ausgeführt wird, erzeugt Commodities ein Ereignis, das dann von ToolManager dazu benutzt wird Programmobjekte zu aktivieren oder Dock-Objekte umzuschalten. Ein Description String hat die folgende Syntax:

```
[<Klasse>] {[<Qualifier>]} [-][upstroke] [<Tastencode>]
```

Alle Befehlsworte können groß oder klein geschrieben werden.

Klasse beschreibt die InputEvent-Klasse. Dieser Parameter ist optional und falls er weggelassen wird, dann wird die Vorgabe rawkey benutzt. Siehe InputEvent-Klassen.

Qualifier sind "Signale", die gesetzt oder nicht gesetzt sein müssen zu dem Zeitpunkt, an dem der Tastenbefehl ausgeführt wird, sonst wird kein Ereignis erzeugt. Für jeden Qualifier, der gesetzt sein soll, müssen Sie das Befehlswort angeben. Alle anderen Qualifier müssen dann nicht gesetzt sein. Falls Sie einen Qualifier ignorieren wollen, dann setzen sie ein - vor sein Befehlswort. Siehe Qualifier.

Normalerweise wird ein Ereignis erzeugt, wenn eine Taste gedrückt wird. Falls das Ereignis generiert werden soll wenn die Taste losgelassen wird, dann müssen Sie das Befehlswort upstroke angeben. Wenn sowohl beim Drücken als auch beim Loslassen der Taste ein Ereignis erzeugt werden soll, dann müssen sie das Befehlswort -upstroke angeben.

Der Tastencode ist abhängig von der InputEvent-Klasse. Siehe Tastencodes.

InputEvent-Klassen
Qualifier
Tastencodes
Beispiele für Tastenbefehle

Achtung: Wählen Sie ihre Tastenbefehle sorgfältig, denn Commodities hat eine hohe Priorität in der InputEvent-Handlerkette, d.h. vorgegebene Definitionen werden übergangen.

1.22 InputEvent-Klassen

InputEvent-Klassen
=====

Commodities unterstützt die meisten der InputEvent-Klassen, die von dem input.device erzeugt werden. Diese Sektion beschreibt die Klassen, die nützlich für ToolManager sind.

rawkey

Dies ist die vorgegebene Klasse. Sie beschreibt alle Ereignisse, die durch die Tastatur erzeugt werden können. Zum Beispiel erzeugt rawkey a oder a jedesmal ein Ereignis, wenn die Taste "a" gedrückt wird. Sie müssen einen Tastencode für diese Klasse angeben. Siehe rawkey.

rawmouse

Diese Klasse beschreibt alle Ereignisse, die durch die Maus erzeugt werden können. Sie müssen einen Tastencode für diese

Klasse angeben. Siehe rawmouse.

diskinserted

Ereignisse dieser Klasse werden generiert, wenn eine Diskette in ein Laufwerk gelegt wird. Diese Klasse besitzt keine Tastenkodes.

diskremoved

Ereignisse dieser Klasse werden generiert, wenn eine Diskette aus einem Laufwerk genommen wird. Diese Klasse besitzt keine Tastenkodes.

1.23 Qualifier

Qualifier

=====

Einige Befehlsworte wurden erst bei Commodities V38 eingeführt. Diese sind mit einem * markiert.

lshift, left_shift *

Linke Shift-Taste.

rshift, right_shift *

Rechte Shift-Taste.

shift

Irgendeine Shift-Taste.

capslock, caps_lock *

Caps-Lock-Taste.

caps

Irgendeine Shift-Taste oder die Caps-Lock-Taste.

control, ctrl *

Control-Taste.

lalt, left_alt *

Linke Alt-Taste.

ralt, right_alt *

Rechte Alt-Taste.

alt

Irgendeine Alt-Taste.

lcommand, lamiga *, left_amiga *, left_command *

Linke Amiga-/Kommando-Taste.

rcommand, ramiga *, right_amiga *, right_command *

Rechte Amiga-/Kommando-Taste.

numericpad, numpad *, num_pad *, numeric_pad *

Dieses Befehlswort muß angegeben werden, wenn eine Taste von der

Zehnertastatur benutzt wird.

leftbutton, lbutton *, left_button *
Linke Maustaste. Siehe unten.

midbutton, mbutton *, middlebutton *, middle_button *
Mittlere Maustaste. Siehe unten.

rbutton, rightbutton *, right_button *
Rechte Maustaste. Siehe unten.

repeat
Dieser Qualifier ist gesetzt, wenn die Tastenwiederholung aktiv ist. Dies ist nur sinnvoll für die InputEvent-Klasse rawkey.

Achtung: Commodities V37 hat einen Fehler, der die Benutzung von leftbutton, midbutton und rbutton als Qualifier verhindert. Dieser Fehler wurde in V38 behoben.

1.24 Tastenkodes

Tastenkodes
=====

Jede InputEvent-Klasse besitzt ihre eigenen Tastenkodes:

rawkey
rawmouse

1.25 rawkey Tastenkodes

Tastenkodes für die InputEvent-Klasse 'rawkey'

Einige Befehlsworte wurden erst bei Commodities V38 eingeführt. Diese sind mit einem * markiert.

a-z, 0-9, ...
ASCII-Zeichen.

f1, f2, ..., f10, f11 *, f12 *
Funktionstasten.

up, cursor_up *, down, cursor_down *
left, cursor_left *, right, cursor_right *
Cursor-Tasten.

esc, escape *, backspace, del, help
tab, comma, return, space, spacebar *

Spezial-Tasten.

enter, insert *, delete *

page_up *, page_down *, home *, end *

Tasten der Zehnertastatur. Diese Tastenkodes müssen mit dem Qualifier numericpad benutzt werden!

1.26 rawmouse Tastenkodes

Tastenkodes für die InputEvent-Klasse 'rawmouse'

Diese Befehlsworte wurden erst bei Commodities V38 eingeführt. Sie sind nicht verfügbar in V37.

mouse_leftpress

Drücke die linke Maustaste.

mouse_middlepress

Drücke die mittlere Maustaste.

mouse_rightpress

Drücke die rechte Maustaste.

Achtung: Um einen dieser Tastenkodes zu benutzen, müssen sie auch das entsprechende Qualifier-Befehlswort angeben, z.B.

rawmouse leftbutton mouse_leftpress

1.27 Beispiele für Tastenbefehle

Beispiele für Tastenbefehle

=====

ralt t

Rechte Alt-Taste festhalten und "t" drücken.

ralt lalt t

Rechte und linke Alt-Taste festhalten und "t" drücken.

alt t

Irgendeine Alt-Taste festhalten und "t" drücken.

rcommand f2

Rechte Amiga-Taste festhalten und die zweite Funktionstaste drücken.

numericpad enter

Enter-Taste auf der Zehnertastatur drücken.

rawmouse midbutton leftbutton mouse_leftpress

Mittlere Maustaste festhalten und die linke Maustaste drücken.

diskinserted

Eine Diskette in ein Laufwerk einlegen.
