

Yak

COLLABORATORS

	<i>TITLE :</i> Yak		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		October 2, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Yak	1
1.1	Documentazione Yak v1.57	1
1.2	Copyright	2
1.3	Nuovi utenti: Nota-bene	2
1.4	Introduzione	3
1.5	Limiti	4
1.6	Lanciare Yak	4
1.7	Cambiare i settaggi di Yak	5
1.8	Attivazioni	6
1.9	AutoPuntamento	7
1.10	Finestre Popup	7
1.11	Attiva RMB	8
1.12	Attiva MMB	8
1.13	Attiva con tasto	8
1.14	Click porta davanti	8
1.15	Click porta dietro	8
1.16	Cicla schermi	9
1.17	Attiva schermo	9
1.18	AmigaDOS *	9
1.19	Niente Click	9
1.20	Patterns d'Inclusione	10
1.21	AutoAttiva Schermi	10
1.22	AutoClicka Schermi	10
1.23	PopWindows	10
1.24	ClickWindows	10
1.25	Settaggi scadenze	11
1.26	Settaggi vari	11
1.27	Volume click	11
1.28	Ritardo AutoPuntamento	11
1.29	Rimpiazza MMB	11

1.30	Scadenza Schermo	11
1.31	Scadenza Mouse	12
1.32	Metodo Spegnimento Mouse	12
1.33	Definizione dei Tasti Rapidi di Yak	12
1.34	Azioni dei Tasti Rapidi	13
1.35	Comando DOS	14
1.36	Chiude Finestra	15
1.37	Zippa Finestra	15
1.38	Minimizza Finestra	15
1.39	Massimizza Finestra	15
1.40	Cicla Finestre	15
1.41	Cicla Indietro Finestre	15
1.42	Apri Palette	16
1.43	Schermo Davanti	16
1.44	Schermo Dietro	16
1.45	Attiva WorkBench	16
1.46	Centra Schermo	17
1.47	Spegne Display	17
1.48	Inserisci Testo	17
1.49	Inserisci Data	18
1.50	Mostra Yak	18
1.51	AmigaDos Patterns	18
1.52	Stringhe per il formato della data	19
1.53	Problemi	20
1.54	Storia del programma	21
1.55	Compilare Yak	22
1.56	Ringraziamenti	22
1.57	Contattare l'autore	23
1.58	Altro sui Tasti Rapidi	23
1.59	Classi di evento	24
1.60	Qualificatori	25
1.61	Codici tasto	26
1.62	Codici tasto per rawkey	26
1.63	Codici tasto per rawmouse	27
1.64	Esempi di Hot Key	27
1.65	Localization	28

Chapter 1

Yak

1.1 Documentazione Yak v1.57

Yak Versione 1.57
di Gaël Marziou
rilasciato il 21 Dicembre 1993.

INDICE DEI CONTENUTI@

Copyright e Distribuzione
Nuovi Utenti: Nota-bene
Introduzione
Limiti
Lanciare Yak
Cambiare i settaggi di Yak
Attivazioni
Patterns
Scadenze
Varie
Tasti Rapidi
Azioni Tasti Rapidi
Altro sui Tasti Rapidi
Patterns dell'AmigaDos
Stringhe per il formato della data
Problemi

Localizzazione
Storia del programma
Compilare Yak
Ringraziamenti
Contattare l'autore

1.2 Copyright

Yak (i binari, i sorgenti e la documentazione) é Copyright © 1992, 1993 Martin W. Scott & Gaël Marziou. All Rights reserved.

Yak é liberamente distribuibile. I sorgenti sono inclusi, e hai il permesso di modificarli per uso personale, ma qualsiasi modifica fatta NON DEVE essere ridistribuita. Se hai effettuato cambiamenti e pensi che ce ne vorrebbero altri, mandameli e li includerò nella versione successiva.

Oltre ciò Yak è gratuito, viene distribuito SENZA GARANZIE. L'autore non é responsabile di alcuna perdita o danneggiamento risultante dall'uso di Yak e perciò l'utente accetta questa responsabilità.

Yak non può essere venduto, può essere copiato al costo nominale di una copia. Non può essere distribuito con un prodotto commerciale senza la precedente autorizzazione dell'autore. Yak deve essere distribuito con tutta la documentazione integra ed inalterata, e preferibilmente con i sorgenti. Implicito é il permesso per Fred Fish di distribuirlo nella sua stupenda raccolta di software di pubblico dominio.

Anche se Yak é gratuito, LE DONAZIONI SARANNO BEN ACCETTE :), sia in soldi che software sviluppato personalmente. Guarda
Contattare l'autore

1.3 Nuovi utenti: Nota-bene

Il formato del file di settaggi di Yak é cambiato di nuovo: é stato diviso
in due files, "S:Yak.prefs" e "S:Yak.hotkeys". Un programma di conversione (Convert) serve proprio a creare i files dei settaggi per la versione 1.5 dalla versione 1.3/1.4. Ci sono due piccole mancanze nella conversione: devi reintrodurre le stringhe di ComandoDOS e di Inserisci Data utilizzando la finestra per la Definizione dei Tasti Rapidi (vedi sezione
Tasti Rapidi
per dettagli su come fare); l'altra piccola mancanza é che un Tasto Rapido
sarà automaticamente creato per la parte "Mostra finestra settaggi", che é probabilmente superfluo, come vedremo in seguito nella spiegazione di CX_POPKEY.

Ci sono alcuni importanti cambiamenti tra la v1.4 e la 1.5 di Yak. Il più importante: la richiesta di stack é cresciuta a 4500 bytes; ricordati di cambiare la dimensione dello stack nell'icona che usi per lanciare Yak.

Secondo, l'icona AppIcon é standard OFF, e dovrebbe essere esplicitamente abilitata utilizzando il tooltype APPICON=TRUE. Questo tooltype rimpiazza il tooltype NOICON.

Terzo, il tooltype CX_POPKEY viene utilizzato per determinare il tasto rapido che apre la finestra dei settaggi di Yak, e questo tasto non può essere alterato o sovrascritto da quello della finestra della definizione dei tasti rapidi (diversamente dalle versioni precedenti); comunque, puoi definire un altro tasto rapido che apre la finestra di Yak.

Quarto, Yak é stato localizzato e prevede ora l'utilizzo dell'inglese come lingua internamente definita e cataloghi italiano, francese e tedesco sono forniti. Se vuoi localizzare Yak nella tua lingua, per favore contattami (vedi sezione

Contattare l'autore
e
Localizzazione
).

Naturalmente, ci sono state molte altre modifiche, e andando girando in questo documento le scoprirai e imparerai ad usare tutte.

1.4 Introduzione

Yak sta per "Yet Another Kommodity" (ancora un'altra "kommodity") é un programma di manipolazione finestre e mouse simile a DMouse, MightyMouse.

Perché scriverne un'altro? Nessuno degli altri (e li ho controllati tutti) mi andavano bene. Yak ha le seguenti caratteristiche:

- o AutoPuntamento (sunmouse) che attiva le finestre e gli schermi solo quando ci si ferma sopra, possono essere esclusi schermi (é compatibile con i programmi di menù popup);
 - o Finestre Popup (le porta davanti) quando sono é selezionato il flag AutoPuntamento;
- o Attiva con tasto le finestre quando premi un tasto su di esse;
- o Attiva RMB/MMB le finestre quando premi il bottone centrale o di destra del mouse;
- o Click per portare finestre davanti o dietro, puoi anche specificare quali finestre/schermi includere/escludere;
- o Cicla gli schermi con il mouse;
- o Spegnimento del Mouse e dello Schermo (con anche un tasto rapido);

- o Funzionalità estensibile delle azioni associate ai tasti rapidi per:
 - Eseguire un comando AmigaDOS;
 - Inserire un testo (con un tasto rapido interna);
 - Inserire la data;
 - Chiudere/Zippare/Minimizzare/Massimizzare le finestre;
 - Ciclare le finestre e gli schermi;
 - Attivare lo schermo del WorkBench;
 - Centrare la posizione dello schermo;
 - Oscurare il display;
 - Cambiare la palette di uno schermo (necessita reqtools);
- o Volume del click dei tasti regolabile;
- o Eliminazione click dei drives;
- o Opzione per avere le wildcard anche con '*' oltre che con '#?';
- o AppIcon opzionale per mostrare l'interfaccia di Yak;
- o Finestra carina per settare i parametri :).

Ti sembrano familiari queste caratteristiche? Yak é una combinazione di AutoPoint, ClickToFront, Blanker e commodity IHelp/FKey con quello che c'è in KCommodity e DMouse.

Tutto questo é abbastanza ridotto: occupa 29K su disco, e circa lo stesso quando é in esecuzione. Utilizza così poca memoria perché fa uso degli overlay; la finestra dei settaggi viene caricata solo quando serve. Questo significa che Yak utilizza più memoria quando apre la finestra di configurazione (circa 45K).

1.5 Limiti

Yak ha solamente un normale programma che oscura lo schermo. Da quando é uscito il 2.0 si sono diffusi parecchi oscura-schermo come Spliner, ASwarm, FracBlank e altri. Ho pensato di includerne uno che semplicemente spegne lo schermo così ho accontentato tutti :).

Non esiste l'accelerazione del mouse. Il sistema di default (settabile con le preferenze Input) é il metodo più adeguato. Se avrò sufficienti richieste implementerò anche una accelerazione.

1.6 Lanciare Yak

Yak é stato sviluppato per essere lanciato dalla directory sys: ←
WBStartup,
ma può essere eseguito da una Shell (non raccomandato, in quanto utilizza più memoria). Tutti i settaggi di Yak sono salvati in 2 files, un file di preferenze "S:Yak.prefs" ed un file sui tasti rapidi "S:Yak.hotkeys". La cosa migliore è utilizzare lo script di installazione quando usi per la

prima volta Yak, lo configuri come vuoi, poi salvi la configurazione per successive esecuzioni.

NB: Yak ha bisogno di una dimensione dello stack di 4500 bytes. Dovresti cambiarlo nell'icona o, se lo esegui da Shell, usare i seguenti comandi:

```
stack 4500
run >nil: yak
```

Gli unici tooltypes che Yak utilizza sono quelli standard del WorkBench (come DONOTWAIT), quelli standard delle commodity (come CX_POPUP), quelli delle AppIcon e quello LANGUAGE. Complessivamente sono i seguenti:

ToolType	Categoria	Descrizione	Default
CX_POPKEY	HOTKEY	Tasto per visualizzare Yak	RCommand Help
CX_PRIORITY	INTEGER	Priorità di questa commodity	0
CX_POPUP	BOOLEAN	Visualizza Yak alla partenza?	NO
APPICON	BOOLEAN	Se TRUE, lancia Yak con icona	FALSE
ICONNAME	STRING	Nome della AppIcon	"Yak!"
ICONXPOS	INTEGER	Coordinata-X della AppIcon	floating
ICONYPOS	INTEGER	Coordinate-Y della AppIcon	floating
LANGUAGE	STRING	Lingua da utilizzare in Yak	Nessuno

Dovresti anche settare il tooltype "DONOTWAIT" se vuoi far partire Yak dalla directory SYS:WBStartup.

La AppIcon é opzionale, e di default é disattivata. Specifica APPICON=TRUE se desideri averla. Se attivi questa opzione, Yak mette la sua AppIcon sullo schermo del WorkBench e, quando clicki due volte, la finestra dei settaggi viene aperta. I colori della AppIcon possono essere cambiati come quelli di una normale icona.

La CX_PRIORITY può essere utile per abilitare Yak e altre commodity in modo che funzionino bene assieme. Guarda, per esempio, in
Attiva RMB/MMB
.

Il tooltype LANGUAGE permette di selezionare, forzandola e se é presente, la lingua di Yak nel seguente modo:

```
LANGUAGE=italiano
```

Naturalmente se questo campo non viene riempito, di default Yak sceglie quello con cui é stato lanciato il WorkBench.

1.7 Cambiare i settaggi di Yak

Tutti i settaggi di Yak possono essere modificati tramite la ↔ finestra dei

settaggi. Una volta che é stato lanciato YAk, puoi portare davanti la finestra con le seguenti procedure:

Premendo RCommand Help (il tasto Destro Amiga e quello di Help).
Questa sequenza di tasti può essere configurato con il tooltype
CX_POPKEY;

Clickando due volte sulla AppIcon (se é presente);

Rilanciando Yak (dall'icona del WorkBench);

Con il programma-commodity Exchange;

La finestra aperta contiene molti gadgets, raccolti in diverse classi.
Queste sono:

Attivazioni

Patterns

Varie

Scadenze

Tasti Rapidi

Ricordati che devi premere RETURN, ENTER o TAB quando hai ←
modificato un

campo stringa, così puoi aggiornare il cambiamento. Clickando fuori del gadget, invece, fa si che la modifica venga persa. (Il tasto TAB attiva il prossimo campo gadget stringa).

Inoltre (e questo si applica in particolare alla finestre dei tasti rapidi) Intuition di default non ti permette di tagliare ed incollare del testo tra gadget stringa, ma se vuoi, puoi utilizzare il programma "NewEdit" di Uwe Roehm, che aggiungono il taglia ed incolla anche a questi campi particolari (si può trovare su un fish disk o in siti ftp).

In aggiunta, i due gadget "Nascondi" e "Esci" effettuano operazioni classiche delle commodity rispettivamente nascondendo la finestra dei settaggi e terminando Yak. Clickando sul gadget di chiusura si effettua una operazione di tipo "Nascondi".

La finestra dei settaggi ha anche voci di menù "Nascondi" ed "Esci" (che fanno le stesse operazioni dei gadget), ed in più voci "Carica settaggi" e "Salva settaggi". Tutte le modifiche dei settaggi di Yak verranno perse se non effettui "Salva settaggi". Per caricare gli ultimi settaggi salvati puoi utilizzare "Carica settaggi".

1.8 Attivazioni

Sono presenti le seguenti attivazioni:

AutoPuntamento

Finestre Popup
Attiva con tasto
Click porta davanti
Click porta dietro
Cicla schermi
Attiva schermo
Attiva MMB
Attiva RMB
AmigaDos *
Niente Click Drive

1.9 AutoPuntamento

Attiva la finestra sotto il mouse. Funziona esattamente con la commodity della Commodore AutoPoint, nel senso che attiva solo una finestra quando ci si ferma su una finestra. Questa funzione é compatibile con le utility dai menù popup come MagicMenu.

Nota che le funzioni AutoPuntamento e Finestre Popup vengono eseguiti solo quando NESSUN qualificatore (del mouse o della tastiera) viene premuto. Questo è stato deciso per evitare azioni non desiderate e per evitare conflitti con altri programmi.

Guarda anche:

Finestre Popup
AutoAttiva Schermi

1.10 Finestre Popup

Solo quando AutoPuntamento é attivo, questa attivazione fa sì che Yak porti davanti le finestre e le attiva. L'eccezione si ha nel caso la finestra sotto il mouse sia di un requester.

Guarda anche:

AutoPuntamento
AutoSchermi
Finestre Popup

1.11 Attiva RMB

Quando selezionato, la finestra sotto il mouse verrà attivata quando saranno premuti il bottone destro, non considerando AutoPuntamento. Questo è utile nel caso in cui si voglia utilizzare un determinato menù e non si voglia aspettare che AutoPuntamento effettui la attivazione.

Per esempio, quando stai usando il gadget di profondità, lo schermo che hai sotto non è attivato, ma con questa attivazione settata, clickando sul tasto destro del mouse avrai i menù corretti.

Nota 1: Se non ci sono window attivate sotto il mouse, la prima finestra dello schermo sarà attivata.

Nota 2: Per funzionare adeguatamente con i programmi di menù popup, è probabile che necessario settare la CX_PRIORITY di Yak maggiore di quella del programma di menù popup.

1.12 Attiva MMB

Quando selezionato, la finestra sotto il mouse sarà attivato quando il tasto centrale verrà premuto senza valutare il valore di AutoPuntamento. Questo è utile nel caso voglia selezionare il menù di una determinata finestra senza attendere la funzione AutoPuntamento o clickare all'interno della finestra.

1.13 Attiva con tasto

Attiva la finestra sotto il mouse quando viene premuto un tasto. Dovresti utilizzare solo una delle attivazioni AutoPuntamento o Attiva con tasto.

1.14 Click porta davanti

Porta davanti la finestra clickando due volte sulla stessa.

Come con AutoPuntamento/Finestre Popup, il click porta davanti e dietro è disabilitato mentre viene premuto un tasto.

Guarda anche:

Schermi Clickabili

Finestre Clickabili

1.15 Click porta dietro

Porta dietro la finestra tenendo premuto il tasto sinistro e \leftrightarrow clickando con quello destro.

Come con AutoPuntamento/Finestre Popup, il click porta davanti e dietro é disabilitato mentre viene premuto un tasto.

Guarda anche:

Schermi Clickabili

1.16 Cicla schermi

Utilizzando la stessa sequenza nella pressione dei tasti della \leftrightarrow funzione Click porta dietro (o Click porta davanti) questa attivazione porta dietro (o davanti) lo schermo attuale se:

- o la finestra sotto il mouse é di tipo backdrop (come quella del WorkBench)
- o c'è solo una finestra sull'attuale schermo.

Guarda anche:

Click porta dietro

Click porta davanti

1.17 Attiva schermo

Quando selezionata, questa funzione automaticamente attiva gli schermi spostandosi tra essi con la funzione Cicla schermi o con LCommand m. L'attivazione é opzionale per evitare conflitti con applicazioni che usano sistemi simili (come WindX di Steve Tibbet).

1.18 AmigaDOS *

Quando abilitato, questo flag permette di utilizzare le wildcard di tipo UNIX "*" oltre che quelle AmigaDos "#?".

1.19 Niente Click

Quando attivato, evita che i drives vuoti clickino.

1.20 Patterns d'Inclusione

I gadget dei pattern ti permettono di specificare quali schermi e finestre possono essere considerate nelle varie funzionalità di Yak. Tutti i pattern si riferiscono ai nomi degli schermi e delle finestre che corrispondono alle wildcard specificate. Tutti i pattern sono standard

```
AmigaDos Patterns
.
```

Sono disponibili i seguenti pattern:

```
AutoAttiva Schermi
AutoClicka Schermi
Finestre PopUp
Finestre Clickabili
```

1.21 AutoAttiva Schermi

```
AutoPuntamento
funzionerà con gli schermi che corrispondono.
```

1.22 AutoClicka Schermi

```
Click porta davanti
e
Click porta dietro
funzionerà sugli schermi
il cui titolo corrisponde al pattern.
```

1.23 PopWindows

```
Attivazione Finestre PopUp
funzionerà con le finestre che corrispondono.
```

1.24 ClickWindows

```
Click porta davanti
funzionerà sulle finestre che corrispondono.
```

1.25 Settaggi scadenze

Questa finestra contiene pochi altri settaggi controllati dai gadget: ←

Tempo Schermo

Tempo Mouse

1.26 Settaggi vari

Yak ha pochi altri controlli da parte di questi gadgets. Sono:

Volume click

Ritardo AutoPuntamento

#

Rimpiazza MMB

Metodo Spegnimento Mouse

1.27 Volume click

Controlla il volume del click dei tasti (il suono che viene emesso quando premi un tasto). Un volume zero significa nessun click (questo sembra ovvio ma con un valore zero l'audio device non viene per niente aperto). Massimo valore 64.

1.28 Ritardo AutoPuntamento

Controlla per quanto tempo bisogna aspettare dopo che il mouse si è fermato prima di attivare la finestra sotto il mouse. Questo valore deve essere tra 0 e 5 secondi e corrisponde a passi di 10ms.

1.29 Rimpiazza MMB

Controlla quali qualificatore MMB (Middle Mouse Button) deve rimpiazzare.

1.30 Scadenza Schermo

Se l'utente non utilizza mouse o tastiera entro il tempo che hai inserito lo schermo verrà oscurato. Mettendo 0 verrà disabilitato del tutto. Lo spegnimento viene effettuato aprendo uno schermo a due colori con la stessa risoluzione di quello davanti.

1.31 Scadenza Mouse

Se il mouse non viene mosso entro il tempo specificato, ←
scomparirà il
puntatore dello stesso. Questa funzione è operativa solamente nel caso che

Metodo Spegnimento Mouse
non sia messo a "Nessuno".

1.32 Metodo Spegnimento Mouse

Questo gadget determina il metodo con il quale il puntatore del ←
mouse deve
essere oscurato. "Nessuno" disabilita lo spegnimento, "Con Sprites" spegne tutti gli sprites, "Con Copper" spegne il puntatore modificando la lista del copper. Questo ultimo metodo risulta comodo per quei programmi che utilizzano per esempio il cursore del testo come sprite, tuttavia è il meno affidabile (qualche il puntatore ritorna) e sembra non funzionare con gli Amiga AGA.

Guarda anche:

Problemi

1.33 Definizione dei Tasti Rapidi di Yak

Clickando sul gadget "Edita Tasti Rapidi" si apre una nuova ←
finestre che
ti permette di creare, editare o cancellare tasti rapidi. Tasto rapido significa un tasto o sequenza di tasti che premuti effettuano una determinata azione (tra quelle che puoi scegliere). I tasti rapidi vengono definiti tramite una stringa di descrizione, che è un modo veramente flessibile di definire eventi di input. Per dettagli sulla creazione di descrizioni di tasti rapidi, guarda
Altro sui Tasti Rapidi
.

Puoi avere quanti tasti rapidi vuoi, ed ogni azione può appartenere a più di un tasto rapido.

Ci sono due liste nella finestra di Definizione Tasti Rapidi; la lista di sinistra indica le azioni disponibili, l'altra i tasti rapidi correntemente definiti per ogni azione.

per aggiungere un nuovo tasto rapido, prima seleziona l'azione che vorresti

utilizzare sullo stesso (clickando sulla lista di sinistra). Poi clicca sul gadget `Aggiungi` nella lista dei Tasti Rapidi. La stringa dei gadget sarà, allora, attiva e potrai inserire esattamente la stringa di descrizione.

Per alcuni tipi di azione, altri gadget diventano attivi. Il gadget `Opzioni` (il gadget che cicla sotto quello delle Azioni) determina quello che accadrà allo schermo quando sarà premuto il Tasto Rapido corrispondente. Attualmente, le opzioni possibili sono quelle di non fare niente, di portare davanti il `WorkBench` o di portare davanti lo schermo pubblico standard. Di solito queste opzioni vengono utilizzate nel tasto rapido `Comando DOS` per mostrare automaticamente lo schermo sul quale si aprirà la finestra.

Il gadget dei Parametri è un gadget stringa che diventa attivo per alcuni tipi di Azioni, e ha come significato quello di inserire una stringa alla esecuzione del Tasto Rapido.

Puoi editare un Tasto Rapido esistente selezionandolo ed editando i gadget relativi. Naturalmente il gadget `Cancella` elimina quel Tasto Rapido dalla lista.

Quando hai finito di editare i Tasti Rapidi, clicca sul gadget `"Ritorna..."` per tornare alla finestra dei settaggi principali, o sul gadget di chiusura per `'nascondere'` Yak. Ricordati di salvare i tuoi cambiamenti utilizzando `"Salva settaggi"` nella finestra principale.

Guarda anche

Azioni dei Tasti Rapidi

1.34 Azioni dei Tasti Rapidi

Le principali azioni disponibili sono:

`Comando DOS`

`Chiude Finestra`

`Zippa Finestra`

`Minimizza Finestra`

`Massimizza Finestra`

`Cicla Finestra`

`Apri Palette`

`Schermo Davanti`

`Schermo Dietro`

`Attiva WorkBench`

`Centra Schermo`

Spegne Display

Inserisce Testo

Inserisce Data

Mostra Yak

Cicla Indietro Finestre

1.35 Comando DOS

Argomento: comando da eseguire

Esegue un comando AmigaDOS come specificato nella stringa dei Parametri. Nota che puoi utilizzare il gadget delle Opzioni per portare il WorkBench o lo schermo pubblico standard davanti (utile nel caso in comando apra una finestra).

Il comando viene eseguito in maniera asincrona, così non ci sono tempi di attesa fintanto che esso parta. Se il comando, inoltre, deve gestire input o output, sarà aperta una finestra di console. Puoi naturalmente utilizzare (come per una shell) i comandi di redirectione.

Esempi:

Un Tasto Rapido per aprire una shell:

Questo viene tradizionalmente collegato al tasto "lcommand esc" ed il mio è settato per eseguire il comando

```
"NewShell CON:79/177/582/78/AmigaShell/CLOSE/ALT2/58/660/197"
```

Nota l'utilizzo del flag ALT nella specifica della console, che è poco documentato (leggi "non menzionato"). Attualmente utilizzo due tasti rapidi: uno per lanciare una shell ed uno per lanciare una CShell.

Un Tasto Rapido per liberare la memoria inutilizzata:

Il SAS/C utilizza le librerie shared che spesso occupano preziosa memoria grafica. Ho un Tasto Rapido per eseguire il comando "avail >nil: flush" che libera questa memoria.

Un Tasto Rapido per visualizzare il contenuto di un disco:

Metti nel Tasto Rapido "diskinserted" e come stringa argomento "Dir df0:".

Guarda anche:

Problemi

Altro sui Tasti Rapidi

1.36 Chiude Finestra

Argomento: NESSUNO

Chiude la corrente finestra attiva (questo è equivalente a clickare sul gadget di chiusura della stessa).

1.37 Zippa Finestra

Argomento: NESSUNO

Zippa (riduce alla dimensione precente) la finestra attiva: equivale a clickare sul gadget "Toggle size" della stessa.

1.38 Minimizza Finestra

Argomento: NESSUNO

Rende la finestra attiva il più piccolo possibile.

1.39 Massimizza Finestra

Argomento: NESSUNO

Rende la finestra attiva il più grande possibile.

1.40 Cicla Finestre

Argomento: NESSUNO

Porta la finestra più indietro davanti. Utile per muoversi tra molte finestre.

1.41 Cicla Indietro Finestre

Argomento: NESSUNO

Funziona al contrario di "Cicla Finestre": manda dietro quella più davanti.

1.42 Apri Palette

Argomento: NESSUNO

Apri una palette (per la scelta dei colori) sullo schermo più davanti. La palette viene lanciata in maniera asincrona cosicché puoi lanciarne quante vuoi (sempre tenendo d'occhio la memoria). Comunque, YAK non può terminare mentre le palette sono aperte. Il gadget Opzioni è abilitato in modo che, se lo desideri, puoi determinare dove aprire la palette.

NOTA: Devi aver installato la `reqtools.library` sul tuo sistema per poter utilizzare questa funzione.

ATTENZIONE: Chiudere sempre la finestra della palette prima di chiudere lo schermo sul quale è stata lanciata, altrimenti probabilmente lo schermo rimarrà aperto e nel caso peggiore avrai un crash del sistema.

1.43 Schermo Davanti

Argomento: NESSUNO

Porta lo schermo più dietro davanti.

Guarda anche:

Attiva Schermo

1.44 Schermo Dietro

Argomento: NESSUNO

Porta lo schermo di fronte dietro a tutti gli altri.

Guarda anche:

Attiva Schermo

1.45 Attiva WorkBench

Argomento: NESSUNO

Attiva una finestra del WorkBench (e se necessario, porta davanti lo schermo del WorkBench). Questo abilita l'accesso ai menù del WorkBench senza dover cercare una finestra del WorkBench da attivare (se, per esempio, hai una finestra di Shell grande come tutto lo schermo).

1.46 Centra Schermo

Argomento: NESSUNO

Centra orizzontalmente lo schermo frontale.

1.47 Spegne Display

Argomento: NESSUNO

Spegne immediatamente il display.

Guarda anche:

Tempo Schermo

.

1.48 Inserisci Testo

Argomento: testo da inserire

Inserisci il testo specificato nella stringa dell'Argomento nel seguente modo:

```
\n    convertito in carattere di a capo
\r    convertito in carattere di a capo
\t    convertito in carattere di tabulazione
\     convertito nel carattere barra inversa \
<descr. tasto> convertito nel tasto rapido specificato
\<    convertito nel carattere <
```

A causa del preprocessing, le stringhe di inserimento possono essere utilizzate per molte cose. Per esempio, ho un tasto rapido per regolare la data e per inserire del testo, utilizzando la stringa di argomento (senza le virgolette):

```
"Oggi è <lcommand d>"
```

Qui, il tasto rapido "lcommand d" è un'altro tasto rapido di Yak che ho regolato per inserire la data. Utilizzando stringhe più complicate puoi creare semplici macro per altri programmi.

AVVERTIMENTO: L'utilizzo di tasti rapidi definiti in maniera multipla andrebbero utilizzati con criterio. Se, per esempio, utilizzi una macro di questo tipo:

```
f1 inserisce testo "<f1>"
f2 inserisce testo "<f2>"
```

Premendo f1 o f2 entrerai in un loop senza fine. Se puoi ancora utilizzare il calcolatore, lancia la Commodity Exchange ed elimina Yak. Poi seleziona la finestra di gestione delle Macro e cancella/ridefinisci il tasto rapido incriminato.

Un'altra stringa della quale stare attenti è quella che chiama altri tasti rapidi (per esempio il tasto di inserimento data visto precedentemente) potrebbe non funzionare come pensi. Supponi che la stringa di argomento sia "<lcommand d>\n". Questo creerebbe un stringa con carattere di a capo e POI con la data, perchè Yak prende la data utilizzando un tasto rapido mentre il carattere di a capo viene mandato subito alla finestra attiva.

1.49 Inserisci Data

Argomento: stringa di formato della data

Inserisce la data nello stream di lettura (e così anche nella corrente finestra attiva). Se stai utilizzando AmigaDos 2.1 o successivi, puoi personalizzare la data da inserire. Questo formato viene specificato nella stringa dell'Argomento. Guarda anche

Stringhe per il formato della data

Se sei sfortunato :) e stai utilizzando AmigaDos 2.0, la data sarà inserita nel formato GG-MMM-AA.

Esempio: il formato "%e %B %Y" genera la data nella forma "1 Maggio 1993".

1.50 Mostra Yak

Argomento: NESSUNO

Mostra la finestra dei settaggi. Questa finestra si aprirà sul corrente schermo se è pubblico altrimenti su quello standard. Questa è la stessa funzione di CX_POPKEY.

1.51 AmigaDos Patterns

I patterns AmigaDos vengono utilizzati per includere/escludere una ←
 lista di
 schermi/finestre con nome per un particolare utilizzo. Questi pattern specificati aumentano la compatibilità con programmi che utilizzano lo stesso sistema.

Il riconoscimento dei pattern è case-sensitive. "Amiga" non è identico ad "AMIGA". I patterns standard AmigaDos sono:

- ? Combacia con un singolo carattere.
- # Combacia l'espressione che segue 0 o più volte.
- (ab|cd) Combacia uno delle due espressioni separate da '|'
- ~ Nega l'espressione che segue. Combacia con tutte le stringhe che non sono quella indicata.
- [abc] Classe di caratteri: combacia con i caratteri della classe.
- % Combacia sempre con 0 caratteri (utile in "(foo|bar|%)").
- * Sinonimo di "#?", non disponibile di standard. Disponibile

```
se
    AmigaDos *
    è stata settata.
```

Se non sei abituato ai patterns, puoi trovare queste espressioni strane. Consulta il tuo manuale di sistema per dettagli più precisi. Ci sono due cose basilari che potresti volere: o una lista di nome determinati che devono essere abilitati, o una lista di nomi che devono essere disabilitati. Per ABILITARE l'operazione su tutti gli oggetti (che possono essere schermi o finestre) utilizza il pattern "#?" (combacia tutto). Per abilitare la operazione su N oggetti di nome da "nome1" a nome "nomeN", utilizza:

```
(nome1|nome2| ... |nomeN)
```

e per DISABILITARE l'operazione per questi nomi, premetti una tilde ~.

```
~(nome1|nome2| ... |nomeN)
```

Un esempio: non voglio che "Finestre Popup" porti davanti le finestre del WorkBench e quelle di Protext (programma della Anorn), così le escludo con il seguente pattern:

```
~(Workbench|#?Arnor#?)
```

Nota che il secondo pattern combacia con un titolo che "Arnor" al suo interno.

Un altro esempio: non voglio l'autoattivazione dello schermo del DirOpus. Siccome non visualizza il titolo dello schermo ho dovuto scovarlo con ARTM e XOper per trovarlo. Attualmente li apre con "DOPUS.1" e numeri successivi per quelli accessori. Ecco il pattern che va utilizzato per escluderli:

```
~(DOPUS#?)
```

Nota 1: Specificare lo schermo del WorkBench nel pattern è un po' complicato siccome nelle varie versione è cambiato. Per la maggior parte delle necessità "#?Workbench#?" dovrebbe andare bene, ma alcune applicazioni cambiano il titolo del Workbench per renderlo significativo rispetto a quella applicazione.

Nota 2: Schermi e finestre senza titolo passano sempre la verifica.

1.52 Stringhe per il formato della data

Per il tasto rapido di inserimento data, devi specificare una stringa in stile localizzato (e perciò hai bisogno di AmigaDos 2.1 o maggiori). Le opzioni disponibili della locale.library sono le seguenti:

```
%a - giorno della settimana abbreviato
%A - giorno della settimana completo
%b - nome del mese abbreviato
%B - nome del mese completo
%c - lo stesso che "%a %b %d %H:%M:%S %Y"
%C - lo stesso che "%a %b %e %T %Z %Y"
```

```

%d - giorno del mese con '0'
%D - lo stesso che "%m/%d/%y"
%e - giorno del mese con spazi
%h - nome del mese abbreviato
%H - ora utilizzando la forma a 24 ore con '0'
%I - ora utilizzando la forma a 12 ore con '0'
%j - data giuliana
%m - numero del mese con '0'
%M - numero dei minuti con '0'
%n - inserisce un carattere di a capo
%p - stringa AM o PM
%q - ora utilizzando la forma 24 ore
%Q - ora utilizzando la forma 12 ore
%r - lo stesso che "%I:%M:%S %p"
%R - lo stesso che "%H:%M"
%S - numero dei secondi con '0'
%t - inserisce un carattere di tabulazione
%T - lo stesso che "%H:%M:%S"
%U - numero della settimana, la Domenica è il primo giorno
%w - numero del giorno della settimana
%W - numero della settimana, il Lunedì è il primo giorno
%x - lo stesso che "%m/%d/%y"
%X - lo stesso che "%H:%M:%S"
%y - anno utilizzando la forma a due cifre e con '0'
%Y - anno utilizzando la forma a quattro cifre e con '0'

```

Questa lista dovrebbe essere esaustiva, e dovrebbe gestire tutto quello che riguarda la data; puoi inserire qualsiasi testo nel tasto rapido della data. Alcuni esempi:

```

"Sono le ore %X" risponde "Sono le ore 22:44:16"
"Buon %A!" risponde "Buon Lunedì!"

```

Se hai bisogno di maggiori dettagli, consulta gli autodoc della locale.library per avere chiarimenti.

1.53 Problemi

Ci sono alcuni problemi che sto attualmente correggendo. Per primo, la shell creata dal tasto rapido Comando Dos non ha lo stack e la corrente directory del momento di boot (nella Startup-Sequence). Mantiene tuttavia il tuo path. Lo stack desiderato e la directory vanno indicate nella Shell-Startup altrimenti saranno rispettivamente di 4000 bytes e la directory SYS:.

Se non ti piace il metodo di spegnimento dello schermo di Yak, puoi provare quello della Commodore che dovrebbe spegnere correttamente tutti gli schermi.

NOTA PER GLI UTENTI DI AMOS: Non mi piace AMOS (questa è una mia opinione) in parte perchè non rispetta il sistema operativo. Inoltre scavalca la gestione di TUTTI i devices di input (come mouse tastiera)... cosicchè lo spegnimento del mouse non viene più ripristinato. Due sono le soluzioni per risolvere i guai che causa AMOS:

- 1) Utilizzare lo spegnimento con il Copper

2) Disabilitare lo spegnimento del mouse mettendo 0 nel Tempo Mouse

1.54 Storia del programma

(* = nuove funzioni)

v1.57 - La finestra principale chiude, prima di terminare, anche le altre eventualmente aperte.

- Compilato con il SAS C 6.5, dimensione del codice ridotta di 2Kb !

v1.56 - La finestra dei settaggi ora si apre sempre sotto il titolo dello schermo.

* Aggiunta la funzione "Attiva MMB".

- Il tasto rapido di chiusura delle finestre ora funziona anche per le finestre di Shell.

* Aggiunta la documentazione in italiano in formato amigaguide.

* Aggiunto il catalogo svedese e danese.

* La finestra dei settaggi ora si apre sul corrente schermo pubblico altrimenti su quello standard.

- Messo a posto un errore che si verificava nel caso in cui si avevano molti schermi visibili. Grazie a Pierre Carette che ha notato questo errore.

* Nuovo significato di Click Porta Davanti, funziona come Click Porta Dietro se "Cicla Schermi" è selezionato.

v1.55 - Corretto uno stupido bug che impediva a Yak 1.54 di funzionare sotto KickStart 2.04.

- Migliorata la gestione dei font con font bassi.

- Script di installazione che ora non sovrascrive sui precedenti per evitare di perdere i tooltypes modificati dall'utente.

- Cambiata la distribuzione dei files sul circuito aminet, i sorgenti e gli eseguibili distinti.

* Aggiunta di nuovo tasto rapido "Cicla indietro finestre"

* Aggiunto catalogo e script di installazione italiano.

v1.54 * Aggiunto il tooltype LANGUAGE che permette di selezionare un linguaggio da quello del WorkBench (descritto il locale.prefs).

* Aggiunta della funzione "Niente Click" per i drives.

* Aggiunto catalogo e script di installazione tedesco.

- Ridotta massicciamente la dimensione dell'eseguibile (10 Kb in meno) utilizzando CATCOMP_BLOCK invece di CATCOMP_ARRAY.

<per la versione 1.53 e precedenti guardare documentazione inglese>

1.55 Compilare Yak

Yak è stato scritto per essere compilato sotto il SAS/C, e dalla versione 1.4a utilizzando il SAS/C 6.2. Il codice passa ottimamente il controllo ANSI cosicché gli utenti non dovrebbero avere problemi a ricompilarlo con altri compilatori ANSI. L'unica cosa da modificare sono le parole chiave specifiche del SAS (come `__saveds`).

Yak utilizza gli overlays, ma questo può essere modificato rimuovendo la definizione `USE_OVERLAYS` in `SCOPTIONS`.

E' stato utilizzato GTB 2.0b per generare l'interfaccia di Yak. La creazione del codice abbisogna solo di piccole modifiche, in particolare nell'utilizzo dei `NewLookMenus` sotto WB3.0 e successivi; i cambiamenti sono indicati dal commento con 4 asterischi, per esempio `/*/*/* ADDED */`.

1.56 Ringraziamenti

Yak è scritto interamente in C e compilato con il SAS/C 6.3. Grazie a Reza Elghazi per l'aiuto nel passaggio alla versione 6.2 ed alla 6.50...

La finestra dei settaggi è stata creata utilizzando l'eccellente (esclusi gli intoppo della v2.0) `GadToolsBox v2.0b`, di Jaba Development.

Yak fa uso dell'`reqtools.library`, che è Copyright di Nico François. Grazie a Matt Dillon per `DMouse` (che mi ha fatto capire molte cose), e a Kai Iske per `KCommodity`, che è il programma dal quale ho 'preso' il suono del click dei tasti. Yak utilizza anche `WB2CLI` un modulo di link di Mike Sinz. Grazie anche a Heinz Wrobel per il suo stupendo porting di RCS su Amiga, a Pierre Carette e Sylvian Rougier per `BrowserII`, Martin Korndörfer per il suo `MagicMenu` e ai miei amici del gruppo Amiga francese che mi hanno aiutato nella localizzazione di Yak.

Grazie a Ingolf Koch, che ha aggiornato la traduzione del catalog effettuata precedentemente da Olaf Gschweng, e ha tradotto in tedesco lo script di installazione e oltre ad alcuni beta testing ! Danke shon ;-)

Grazie ad Alessandro Galassi che ha effettuato la traduzione in italiano del catalogo, dello script di installazione e della documentazione in formato `amigaguide` ! Grazie ;-)

Grazie a Johan Billing che ha effettuato la traduzione del catalog svedese.

Grazie a Patrick van Beem che ha effettuato la traduzione del catalog in danese.

La documentazione sull'utilizzo dei Tasti Rapidi è stata presa dal ToolsManager per gentile concessione di Stefan Becker.

Grazie a Stefan Sticht per la sua commodity PD MouseBlanker dalla quale ho tratto il metodo di spegnimento del mouse con il 'Copper'.

Un grand e grazie a tutti quelli che mi hanno scritto riguardo Yak e mi hanno suggerito modifiche ed indicato problemi.

Grazie anche a Martin Huttenloher che ha creato le icone per le directory e per i files amigaguide che io ho preso dal suo pacchetto MagicWB.

Per ultimo e non meno importante, grazie a Martin Scott che ha creato Yak.

1.57 Contattare l'autore

L'autore può essere raggiunto con commenti, suggerimenti, riferimento di problemi, preghiere, soldi, Amiga 5000 :), etc. al seguente indirizzo:

Gaël Marziou
Cidex 103
38920 CROLLES
FRANCE

O VIA EMAIL (preferito): gael@gnlab030.grenoble.hp.com

1.58 Altro sui Tasti Rapidi

Come definire un Hot Key

Questo capitolo descrive come definire un Hot Key come una stringa di descrizione dell'input che verrà poi interpretata dalle Commodities Exchange. Ogni volta che viene attivato un Hot Key le Commodities genereranno un evento che è utilizzato da ToolManager per attivare gli oggetti Programma oppure per chiudere o aprire un oggetto Dock. Una stringa di descrizione ha la seguente sintassi:

```
[<classe>] {[-][<qualificatore>]} [-][upstroke] [<codice tasto>]
```

Tutte le parole chiave possono essere scritte sia in minuscolo che in maiuscolo.

classe descrive una classe di evento di input (InputEvent). Questo parametro è opzionale e se non viene specificato verrà utilizzato il valore di default rawkey. Vedi

Classi di evento

.

I qualificatori sono "segnali" che devono essere impostati o meno nel momento in cui il tasto Hot Key è attivato; in caso contrario non verrà generato alcun evento. Per ogni qualificatore che deve essere impostato dovete fornire la sua parola chiave. Se volete ignorare un qualificatore basta far precedere la sua parola chiave dal segno -. Vedi

Qualificatori

.

Normalmente un evento di Hot Key viene generato alla pressione di un tasto. Se l'evento deve essere generato invece quando il tasto è rilasciato (anziché premuto) specificate la parola chiave upstroke. Quando sia la pressione che il rilascio di un tasto devono generare un evento specificate -upstroke.

Il codice tasto è dipendente dalla classe di evento. Vedi

Codici tasto

.

Classi di evento

Qualificatori

Codici tasto

Esempi di Hot Key

Nota: scegliete con cura i vostri Hot Key poiché le Commodities hanno una priorità alta nella catena dei gestori di input (in altre parole, gli Hot Key da voi definiti saranno sovrapposti ad altri precedentemente definiti).

1.59 Classi di evento

Classi di evento

=====

Le Commodities supportano molte delle classi di evento (InputEvent) generate dall'input.device. Questa sezione descrive quelle classi che sono più utili per gli Hot Key di ToolManager.

rawkey

Questa è la classe di default e ricopre tutti gli eventi di tastiera. Per esempio rawkey a o a genera un evento ogni volta che viene premuto il tasto "a". Dovete specificare un codice tasto per questa classe. Vedi

rawkey

.

rawmouse

Questa classe descrive tutti gli eventi relativi ai bottoni del

mouse. Dovete specificare un codice tasto per questa classe. Vedi

```
rawmouse
.
```

diskinserted

Gli eventi di questa classe sono generati quando un disco viene inserito in un drive. Questa classe non ha alcun codice tasto.

diskremoved

Gli eventi di questa classe sono generati quando un disco viene estratto da un drive. Questa classe non ha alcun codice tasto.

1.60 Qualificatori

Qualificatori
=====

Alcuni sinonimi delle parole chiave sono stati aggiunti nelle Commodities V38. Questi sono evidenziati da *.

lshift, left_shift *
Tasto Shift sinistro.

rshift, right_shift *
Tasto Shift destro.

shift
Uno dei due tasti Shift.

capslock, caps_lock *
Tasto Caps Lock.

caps
Uno dei due tasti Shift o il tasto Caps Lock.

control, ctrl *
Tasto Control.

lalt, left_alt *
Tasto Alt sinistro.

ralt, right_alt *
Tasto Alt destro.

alt
Uno dei due tasti Alt.

lcommand, lamiga *, left_amiga *, left_command *
Tasto Amiga/Command sinistro.

rcommand, ramiga *, right_amiga *, right_command *
Tasto Amiga/Command destro.

numericpad, numpad *, num_pad *, numeric_pad *

Questa parola chiave deve essere usata per ogni tasto presente sul tastierino numerico.

leftbutton, lbutton *, left_button *

Tasto mouse sinistro. Vedi note più sotto.

midbutton, mbutton *, middlebutton *, middle_button *

Tasto mouse centrale. Vedi note più sotto.

rbutton, rightbutton *, right_button *

Tasto mouse destro. Vedi note più sotto.

repeat

Questo qualificatore è impostato quando è attiva la ripetizione dei tasti. Utile solo nei casi di classe di evento rawkey.

Nota: le Commodities V37 hanno un bug che inibisce l'uso dei qualificatori leftbutton, midbutton e rbutton. Questo bug è stato corretto nella V38.

1.61 Codici tasto

Codici tasto

=====

Ogni classe di evento (InputEvent) ha i suoi codici tasto:

rawkey

rawmouse

1.62 Codici tasto per rawkey

Codici tasto per la classe di evento 'rawkey'

Alcuni sinonimi delle parole chiave sono stati aggiunti nelle Commodities V38. Questi sono evidenziati da *.

a-z, 0-9, ...

Caratteri ASCII.

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

Tasti funzione.

up, cursor_up *, down, cursor_down *

```
left, cursor_left *, right, cursor_right *
  Tasti cursore.
```

```
esc, escape *, backspace, del, help
tab, comma, return, space, spacebar *
  Tasti speciali.
```

```
enter, insert *, delete *
page_up *, page_down *, home *, end *
  Tasti del tastierino numerico. Ognuno di questi codici tasto deve
  essere usato con il qualificatore numericpad!
```

1.63 Codici tasto per rawmouse

Codici tasto per la classe di evento 'rawmouse'

Queste parole chiave sono state aggiunte nelle Commodities V38 e non sono pertanto disponibili nella V37.

```
mouse_leftpress
  Pressione del tasto sinistro del mouse.
```

```
mouse_middlepress
  Pressione del tasto centrale del mouse.
```

```
mouse_rightpress
  Pressione del tasto destro del mouse.
```

Nota: per usare uno di questi codici tasto dovete anche specificare il corrispondente qualificatore, per esempio:

```
rawmouse leftbutton mouse_leftpress
```

1.64 Esempi di Hot Key

Esempi di Hot Key

=====

```
ralt t
  Premere contemporaneamente il tasto Alt destro e il tasto "t"
```

```
ralt lalt t
  Premere contemporaneamente i tasti Alt sinistro e destro e il tasto
  "t"
```

```
alt t
  Premere contemporaneamente uno dei tasti Alt e il tasto "t"
```

```
rcommand f2
```

Premere contemporaneamente il tasto Amiga destro e il secondo tasto funzione

numericpad enter

Premere il tasto Enter sul tastierino numerico

rawmouse midbutton leftbutton mouse_leftpress

Premere contemporaneamente i tasti sinistro e centrale del mouse

diskinserted

Inserire un disco in un drive

1.65 Localization

Yak è ora localizzato. Attualmente supporta l'inglese come linguaggio interno e sono forniti il catalogo italiano, francese e tedesco. Se vuoi localizzare Yak nel tuo linguaggio sei il benvenuto. Non c'è bisogno di essere sviluppatori per fare la traduzione, l'unico requisito è conoscere bene la propria lingua :).

Le cose che possono essere tradotte sono molte:

Yak stesso scrivendo il catalogo.

Lo script di installazione.

La documentazione amigaguide.

Puoi tradurre uno di questi o tutti.

Nella directory Catalogs, troverai un file di nome yak.ct che contiene le stringhe utilizzate dall'interfaccia di Yak e che sono state inserite internamente da Yak.cd che pure troverai nella directory Catalogs.

Ecco un esempio:

```
MISCELLANEOUS_STRING
```

```
; Miscellaneous;
```

Se vuoi tradurre "Miscellaneous" in italiano, dovresti mettere la traduzione nella riga vuota così:

```
MISCELLANEOUS_STRING
```

```
Varie
```

```
; Miscellaneous;
```

Così, quando hai tradotto tutte le stringhe di yak.ct puoi utilizzare (se sei sviluppatore) CatComp o FlexCat (di pubblico dominio) per generare tu stesso il catalogo o mandarmi il file yak.ct cosicché lo tradurro io stesso e lo distribuirò nel pacchetto.

Ci sono 5 linee di ringraziamento in Yak riservate al traduttore, non esitare a mettere il tuo nome ed il tuo indirizzo.
