

Yak

| |
|----------------------|
| COLLABORATORS |
|----------------------|

| | <i>TITLE :</i> Yak | | |
|---------------|-----------------------|---------------|------------------|
| <i>ACTION</i> | <i>NAME</i> | <i>DATE</i> | <i>SIGNATURE</i> |
| WRITTEN BY | | July 27, 2024 | |

| |
|-------------------------|
| REVISION HISTORY |
|-------------------------|

| NUMBER | DATE | DESCRIPTION | NAME |
|--------|------|-------------|------|
| | | | |

Contents

| | | |
|----------|--|----------|
| 1 | Yak | 1 |
| 1.1 | Documentation de Yak v1.52 | 1 |
| 1.2 | Copyright | 1 |
| 1.3 | Utilisateurs des versions précédentes : Notez-bien | 2 |
| 1.4 | Introduction | 2 |
| 1.5 | Humour | 4 |
| 1.6 | Limitations | 4 |
| 1.7 | Démarrer Yak | 4 |
| 1.8 | Changing Yak's settings | 5 |
| 1.9 | Sélecteurs (Toggles) | 6 |
| 1.10 | AutoPointage | 6 |
| 1.11 | Fenêtres PopUp | 7 |
| 1.12 | Activation par le RMB (bouton droit de la souris) | 7 |
| 1.13 | Activation au clavier | 7 |
| 1.14 | Cliquer en avant | 7 |
| 1.15 | Cliquer en arrière | 8 |
| 1.16 | Ecran en arrière | 8 |
| 1.17 | Activer écran | 8 |
| 1.18 | Joker AmigaDOS | 8 |
| 1.19 | Motifs d'inclusion | 8 |
| 1.20 | Ecrans AutoPointables | 9 |
| 1.21 | Ecrans Cliquables | 9 |
| 1.22 | Fenêtres Poppables | 9 |
| 1.23 | Fenêtres Cliquables | 9 |
| 1.24 | Divers | 9 |
| 1.25 | Délais d'extinction | 9 |
| 1.26 | Volume clavier | 10 |
| 1.27 | Délais d'extinction de l'écran | 10 |
| 1.28 | Délais d'extinction de la souris | 10 |
| 1.29 | Masquer la souris | 10 |

| | | |
|------|---|----|
| 1.30 | Touches d'appel de Yak | 10 |
| 1.31 | Actions des Hotkeys | 11 |
| 1.32 | Command DOS | 12 |
| 1.33 | Fermer la fenêtre | 12 |
| 1.34 | Zipper la fenêtre | 12 |
| 1.35 | Minimiser la fenêtre | 13 |
| 1.36 | Maximiser la fenêtre | 13 |
| 1.37 | Cycler les fenêtres | 13 |
| 1.38 | Ouvrir une palette | 13 |
| 1.39 | Ecran en avant | 14 |
| 1.40 | Ecran en arrière | 14 |
| 1.41 | Activer le Workbench | 14 |
| 1.42 | Centrer l'écran | 14 |
| 1.43 | Eteindre l'écran | 14 |
| 1.44 | Insérer du texte | 14 |
| 1.45 | Insérer la date | 15 |
| 1.46 | Configurer Yak | 16 |
| 1.47 | Motifs AmigaDos | 16 |
| 1.48 | Spécifier un format de date | 17 |
| 1.49 | Problèmes | 18 |
| 1.50 | Historique du programme | 19 |
| 1.51 | Compiler Yak | 23 |
| 1.52 | Remerciements | 23 |
| 1.53 | Contacteur l'auteur | 24 |
| 1.54 | Détails sur les touches d'appel | 24 |

Chapter 1

Yak

1.1 Documentation de Yak v1.52

Yak Version 1.52
par Gaël Marziou

CONTENU

Copyright et Distribution
Utilisateurs des versions précédentes : Notez-bien
Introduction
Limitations
Démarrer Yak
Changer la configuration de Yak
Options actives Motifs d'inclusion Divers
Délais d'extinction Touches d'appel
Actions des touches d'appel
Patterns AmigaDOS
Spécifier un format de date
Problèmes
Historique du programme
Compiler Yak
Remerciements
Contacter l'auteur
Détails sur les touches d'appel

1.2 Copyright

Yak (le binaire, les sources et la documentation) est
Copyright © 1992, 1993 Martin W. Scott et Gaël Marziou. Tout droits réservés.

Yak est librement redistribuable. Le source est inclus, et vous êtes autorisé à le modifier pour votre usage personnel, mais aucune des modifications faites ne doit être distribuée. Si vous avez fait des changements que vous pensez pouvoir être utiles à d'autres, envoyez les moi et je les inclurai dans une version future.

Comme Yak est gratuit, il n'est livré avec AUCUNE GARANTIE. L'auteur n'est pas responsable d'une quelconque perte ou dommage pouvant survenir durant l'utilisation de Yak. L'utilisateur prend seul cette responsabilité.

Aucune charge financière ne doit être appliquée à Yak, à part pour couvrir des frais de duplication. Yak ne peut pas être distribué avec un produit commercial sans le consentement de l'auteur. Yak doit être distribué avec toute sa documentation intacte et non altérée, et de préférence avec le source. La permission de distribuer Yak dans sa superbe collection de disquettes est expressément donnée à Fred Fish.

Voir Contacter l'auteur .

1.3 Utilisateurs des versions précédentes : Notez-bien

Le format du fichier de préférence de Yak a encore changé, les préférences étant maintenant séparées en deux fichiers "S:Yak.prefs" et "S:Yak.hotkeys". Le programme Convert est fourni pour créer les fichiers de préférences 1.5 à partir d'un fichier de préférences 1.3/1.4. Il y a deux problèmes dans la conversion, vous devez retaper les définitions de PopCommand et DateFormat en utilisant la nouvelle fenêtre de définition des touches d'appel (voir section Touches d'appel pour plus de détails).

L'autre problème est qu'une touche d'appel sera créée pour la fonction "Configurer Yak", ce qui est probablement superflu étant la discussion ci-dessous sur CX_POPKEY.

Il y a des changements importants entre les versions 1.4 et 1.5 de Yak. Le plus important est que la taille de la pile nécessaire à Yak a été augmentée à 4500 octets, donc pensez à changer la taille de la pile dans l'icône que vous utilisez pour lancer Yak..

Deuxièmement, l'AppIcon n'est pas mise par défaut, and doit être explicitement activée en utilisant le type d'outil APPICON=TRUE dans l'icône de Yak. Ce type d'outil remplace le précédent NOICON.

Troisièmement, le type d'outil CX_POPKEY est utilisé pour déterminer la touche d'appel permettant d'ouvrir la fenêtre de configuration de Yak, cette définition ne peut être altérée ou supplantée depuis la fenetre de définition des touches d'appel (contrairement aux versions précédentes); quoiqu'il en soit, vous pouvez créer d'autres touches d'appel qui ouvrent aussi la fenêtre de Yak.

Bien sûr, il y a eu beaucoup d'autres changements, et vous devriez parcourir ce document afin de vous accoutumer à ceux-ci.

1.4 Introduction

Yak est l'abréviation de "Yet Another Kommodity" (en anglais : encore un autre utilitaire) (je n'ai jamais été bon en orthographe - voir Humour), et est un programme de gestion de fenetres a la souris dans la lignee de DMouse, MightyMouse etc.

Pourquoi en écrire un autre? Aucun des autres (et je les ai pourtant pratiquement tous essayés) ne me convenait.

Yak a les fonctions suivantes :

- o AutoPointage (sunmouse) qui active une fenêtre seulement quand la souris s'arrête dessus; possibilité de spécifier à quels écrans appliquer ou non cette fonctionnalité (Compatible avec les programmes du type PopUpMenu)
- o Fenêtres PopUp se mettent à l'avant plan dès qu'elles sont auto-activées.
- o Activation de la fenêtre quand une touche est pressée (Activation au clavier)
- o Cliquer sur les fenetres pour les faire passer à l'avant ou l'arrière plan ; là aussi possibilité de spécifier à quels écrans appliquer ou non cette fonctionnalité
- o Faire cycler les écrans avec la souris
- o éteindre l'écran et la souris (par touche d'appel aussi)
- o Système extensible de touches d'appel (comme celui de FKey) avec des actions pour:

Exécuter une Command Dos

Insérer un texte (avec des touches d'appel imbriquées)

Insérer la date (à votre format si la locale est présente)

Fermer/Zipper/Minimiser/Maximiser des fenêtres

Cycler des écrans et des fenêtres

Activer le Workbench

Centrer l'écran à l'avant plan

Eteindre l'affichage

Ouvrir une palette sur l'écran à l'avant plan

(nécessite la reqtools.library)

- o Produit un "clic" dont le volume est ajustable à chaque appui sur le clavier;
- o Option "Joker AmigaDos" (comme StarBurst) qui permet d'utiliser '*' comme joker de l'amigaDOS.
- o AppIcon optionnelle sur l'ecran du workbench pour ouvrir la fenêtre des préférences.
- o Jolie fenêtre de configuration utilisant la librairie gadtools;
- o Interface localisée.

Ca vous semble familier? C'est une combinaison des utilitaires standards AutoPoint, ClickToFront, Blanker et IHelp (sur la disquette Extras) plus un soupçon de KCommodity et DMouse.

Et malgré tout, la taille de l'exécutable est faible: environ 30 Ko sur disque, et à peu près la même chose en mémoire en utilisation courante et

environ 54 Ko en mémoire quand on ouvre la fenêtre de configuration. Cette gestion efficace de la mémoire est due à l'utilisation d'overlays (code se chargeant en plusieurs fois) le code de la fenêtre de configuration n'étant chargé que lorsque c'est nécessaire.

1.5 Humour

Ces blagues sont plutôt vieilles mais toujours aussi bonnes:

Q. What's the difference between a goldfish and a goat?

A. A goldfish mucks around in fountains!

Et si vous avez compris celle-ci, vous comprendrez surement celle-là ...

Q. What's the difference between a magic wand and a truncheon (baton).

A. A magic wand is used for cunning stunts.

(Les gens sèchent *toujours* sur la dernière)

Note du traducteur: J'ai préféré ne pas essayer de traduire ces jeux de mots et contrepétteries anglais bien que Martin ait pris le temps de me les expliquer.

1.6 Limitations

L'économiseur d'écran (screen blanker) de Yak est rudimentaire, il éteint simplement l'écran. Depuis l'introduction du 2.0, il est apparu une profusion de screen blankers plus esthétiques, comme Spliner (que j'utilise), ASwarm (plusieurs versions), SuperDark et d'autres. Je pense que la plupart des gens ont chacun leur screen blanker évolué favori, c'est pourquoi je n'en ai pas incorporé dans Yak.

Il n'y a pas d'accélérateur de souris. Celui du système (configurable par les préférences Input) est plus qu'adéquat en ce qui me concerne. Si je reçois suffisamment de demandes, j'ajouterai un accélérateur de souris (pour l'instant je n'en ai pas encore reçues).

1.7 Démarrer Yak

Yak a été conçu pour être lancé depuis le tiroir sys:WBStartup, mais peut être lancé depuis un shell en tapant "run Yak". Tous les paramètres de configuration de Yak sont sauvés dans deux fichiers de préférences "S:Yak.prefs" et un fichier pour les touches d'appel, "S:Yak.hotkeys". La première fois que vous utilisez Yak, configurez le comme vous le souhaitez, puis sauvez cette configuration pour les utilisations suivantes.

NB : Yak a besoin d'une pile de 4500 bytes. Vous devriez le spécifier dans l'icône de Yak, ou si vous le lancez depuis le shell, utilisez les commandes suivantes:

```
stack 4500
run >nil: Yak
```

Les seuls types d'outil que Yak prend en compte sont ceux qui sont standards au Workbench (tel que DONOTWAIT), aux utilitaires (commodities) (tel que CX_POPUP) et ceux des "AppIcon". Ces types d'outil sont :

| Type d'outil | Catégorie | Description | Default (1) |
|--------------|-----------|--|---------------|
| CX_POPKEY | HOTKEY | touche d'appel de la fenêtre de configuration | RCommand Help |
| CX_PRIORITY | ENTIER | Priorité de Yak | 0 |
| CX_POPUP | BOOLEEN | Ouvre la fenetre de configuration au lancement | NO |
| APPICON | BOOLEEN | S'il est à FALSE, pas d'appIcon | FALSE |
| ICONNAME | TEXTE | Nom de l'AppIcon | "Yak!" |
| ICONXPOS | ENTIER | coordonnée en x de l'AppIcon | floating |
| ICONYPOS | ENTIER | coordonnée en y de l'AppIcon | floating |

Vous devriez aussi ajouter le type d'outil "DONOTWAIT" dans l'icône de Yak si vous souhaitez lancer Yak depuis votre tiroir SYS:WBStartup drawer.

La fonction AppIcon est optionnelle et par défaut n'est pas sélectionnée (spécifiez APPICON=TRUE si vous la voulez). Dans ce cas, Yak met une AppIcon sur l'écran du Workbench, et quand vous double-cliquez dessus, la fenêtre de configuration s'ouvre. L'image de l'icone est obtenue en recopiant l'icone de Yak au moment de son lancement ce qui vous permet de personnaliser votre AppIcon pour l'adapter a vos couleurs ou votre résolution simplement en changeant l'icône de Yak.

CX_PRIORITY peut être utile pour permettre un fonctionnement harmonieux de Yak avec d'autres utilitaires (Commodities). Voir, par exemple les notes sur l'option Activation par le RMB

1.8 Changing Yak's settings

Tous les paramètres de configuration de Yak peuvent être modifiés au moyen de sa fenêtre de configuration. Une fois que Yak est lancé, vous pouvez faire apparaître cette fenêtre de ces différentes manières :

En pressant RCommand Help (la touche Amiga de droite et la touche Help). Cette combinaison est configurable

En double-cliquant l'AppIcon de Yak (si cette option a été choisie)

En lançant Yak à nouveau (depuis son icône sur votre disque).

Grace au programme Commodities Exchange (sur la disquette "Extras")

La fenêtre ainsi ouverte contient beaucoup de gadgets, regroupés en classes. Ces classes sont:

Options actives

Motifs d'inclusion
Divers
Délais d'extinction
Touches d'appel

Souvenez vous que vous devez toujours appuyer sur la touche RETURN, ENTER ou TAB une fois que vous avez édité un string gadget, pour que le changement soit enregistré. Vous pouvez perdre vos changements simplement en cliquant à l'extérieur du gadget. (la touche TAB active le prochain string gadget)

Aussi et ceci s'applique particulièrement à la fenêtre des touches d'appel), Intuition par défaut ne vous permet pas de faire du copier/coller entre 2 string gadgets, si néanmoins vous souhaitez en bénéficier, vous devriez obtenir "NewEdit" d'Uwe Roehm, qui leur ajoute un copier/coller par le clipboard (disponible sur une disquette Fish ou sur des sites ftp).

Finalement, les 2 gadgets "Cacher la fenêtre" et "Quitter" correspondent aux fonctions standards des utilitaires consistant respectivement à cacher la fenêtre de configuration et à stopper Yak. Cliquer sur le gadget de fermeture de la fenêtre est équivalent à cacher la fenêtre et non à stopper Yak.

La fenêtre de configuration possède aussi un menu avec des items "Cacher la fenêtre" et "Quitter" (qui opèrent à l'instar des gadgets du même nom), plus les items "Charger les Préfs" et "Sauver les Préfs". Tous les changements de configuration de Yak seront perdus à moins que vous ne sélectionniez l'item "Sauver les Préfs". Si vous voulez retrouver la dernière configuration sauvee utilisez l'item "Charger les Préfs".

1.9 Sélecteurs (Toggles)

Les sélecteurs (toggles) suivants sont disponibles:

AutoPointage
Fenêtres PopUp
Activation au clavier
Cliquer en avant
Cliquer en arrière
Ecran en arrière
Activer écran
Activation par le RMB
Joker AmigaDOS

1.10 AutoPointage

Active la fenêtre sous la souris. Se comporte presque comme l'utilitaire AutoPoint de Commodore, à part qu'il n'active une fenetre que lorsque la souris s'arrete (d'où un petit délai).AutoPoint est compatible avec les programmes de type popup-menu tels que l'excellent MagicMenu.

Il est à noter que les fonctions "AutoPointage" et "Fenêtres PopUp" ne s'appliquent que lorsqu'aucun qualifieur (souris ou clavier) n'est pressé. Non seulement cela évite des conflits avec d'autres programmes, mais en plus cela fournit un

moyen de prévenir certaines activations ou mises en avant plan indésirables.

Voir aussi: Fenêtres PopUp Ecrans AutoPointables

1.11 Fenêtres PopUp

Opère seulement quand AutoPointage est sélectionné, cela indique à Yak qu'il doit amener les fenêtres à l'avant plan en même temps qu'il les active. Il y a une exception à cette règle quand la fenêtre sous la souris a un requester d'ouvert.

Voir aussi: AutoPoint AutoScreens PopWindows

1.12 Activation par le RMB (bouton droit de la souris)

Quand ce mode est sélectionné, la fenêtre sous la souris sera activée quand le bouton droit de la souris est enfoncé, qu'AutoPoint soit sélectionné ou non. Ceci est très utile pour activer le menu qui vous interesse sans avoir soit à attendre qu'AutoPint ait activée la fenêtre soit à cliquer dans la feêtre pour l'activer.

Par exemple, en utilisant les gadgets de profondeur des écrans (coin supérieur droit) le nouvel écran à l'avant-plan n'est pas activé, mais avec "Activation par RMB" cliquer sur le bouton droit de la souris fera apparaître les menus corrects.

Note 1: S'il n'y a pas de fenêtre sous la souris, la première fenêtre de l'écran sera activée.

Note 2: Pour fonctionner correctement avec les programmes de type popupmenu, la priorité de Yak fixée par le type d'outil CX_PRIORITY doit être supérieure à la priorité du programme popupmenu.

1.13 Activation au clavier

Active la fenetre sous la souris quand une touche est pressée.

Vous ne devriez avoir besoin que l'une de ces 2 fonctions "AutoPointage" et "Activation au clavier".

1.14 Cliquer en avant

Amène une fenêtre à l'avant plan en double cliquant dessus.

Comme pour les fonctions "AutoPointage" et "Fenêtres PopUp", cette fonction est ineffective quand un qualifier clavier est enfoncé (touches ALT, SHIFT, CTRL et AMIGA).

Voir aussi: Ecrans cliquables Fenêtres Cliquables

1.15 Cliquer en arrière

Envoye une fenêtre à l'arrière plan en cliquant et maintenant le bouton gauche de la souris, puis en cliquant le bouton droit de la souris.

Comme pour les fonctions "AutoPointage" et "Fenêtres PopUp", cette fonction est ineffective quand un qualifier clavier est enfoncé (touches ALT, SHIFT, CTRL et AMIGA).

Voir aussi: Ecrans Cliquables

1.16 Ecran en arrière

En utilisant la même séquence que "Cliquer en avant" "Cliquer en arrière", vous envoyez un écran à l'arrière plan si:

```
soit la fenêtre sous la souris est une fenetre de fond (comme la
    fenêtre principale du Workbench)
soit il n'y a qu'une fenêtre sur cet écran.
```

Voir aussi: Cliquer en arrière

1.17 Activer écran

Quand il est sélectionné, Yak activera les écrans qu'il manipule par hotkey (c'est-à-dire par les touches d'appel "Cycler les fenêtres" et LCommand m). Ceci est optionnel car il peut y avoir des conflits avec certains programmes. Ceci est similaire (mais pas identique) à WindX de Steve Tibbet. Par activer un écran, j'entend qu'il active la fenêtre sous la souris dans le nouvel écran à l'avant plan.

1.18 Joker AmigaDOS

Quand il est sélectionné, permet l'utilisation de * en tant que caractère de "pattern-matching" (comme MSDOS et UNIX). (C'est ce que fait l'utilitaire StarBurst.)

1.19 Motifs d'inclusion

Les gadgets de la classe "Motifs d'inclusion" vous permettent de spécifier quels écrans et fenêtres sont affectés par diverses fonctions de Yak. C'est-à-dire que le titre de la fenêtre ou de l'écran doit FAIRE PARTIE du motif pour que la fonction correspondante s'applique. Tous les motifs d'inclusion sont des Motifs (Patterns) AmigaDos standards.

Les patterns suivantes sont disponibles:

```
Ecrans AutoPointables  
Ecrans Cliquables  
Fenêtres Poppables  
Fenêtres Cliquables
```

1.20 Ecrans AutoPointables

La fonction `AutoPointage` ne fonctionnera que pour les écrans dont le titre correspond à ce motif.

1.21 Ecrans Cliquables

Les fonctions `Cliquer en avant` et `Cliquer en arrière` ne fonctionneront que pour les écrans dont le titre correspond à ce motif..

1.22 Fenêtres Poppables

La fonction `Fenêtres PopUp` ne fonctionnera que pour les fenêtres dont le titre correspond à ce motif.

1.23 Fenêtres Cliquables

La fonction `Cliquer en avant` ne fonctionnera que pour les fenêtres dont le titre correspond à ce motif.

1.24 Divers

Yak possède quelques autres fonctions contrôlées par les gadgets regroupés sous la classe `'Divers'`.

Ce sont :

```
Volume clavier  
Masquer la souris
```

1.25 Délais d'extinction

Sont regroupés ici, deux gadgets permettant de fixer le délais en secondes d'extinction de :

```
Ecran  
Souris
```

1.26 Volume clavier

Contrôle le volume sonore du clic du clavier. Un volume égal à zéro signifie 'pas de clic' (oui, je sais c'est évident, mais quand il est mis à zéro, le device audio ne sera pas ouvert du tout). Le volume maximum est 64.

1.27 Délais d'extinction de l'écran

Si aucune entrée au clavier ou à la souris n'intervient durant cette durée (en secondes), l'affichage s'éteindra. Mettez cette valeur à zéro pour ne pas utiliser cette fonction. L'extinction (blanking) est obtenue en ouvrant un écran de 2 couleurs dans le même mode d'affichage que l'écran à l'avant-plan.

1.28 Délais d'extinction de la souris

Si la souris n'est pas déplacée pendant ce temps configurable, le pointeur de la souris disparaît. Ceci n'est opérationnel que si l'option Masquer la souris est effective.

1.29 Masquer la souris

Ce gadget détermine quelle méthode utiliser pour faire disparaître la souris (fonction mouse blanking) . 'Non' signifie que cette fonction est désactivée, 'par les Sprites' signifie que Yak fait disparaître le pointeur de la souris en désactivant tous les sprites, et 'par le Copper' signifie que cela est obtenu en modifiant la copper list. Cette dernière option ne désactive que le sprite 0 (le pointeur de la souris), si bien que les émulateurs de terminaux qui utilisent un sprite pour dessiner le curseur fonctionneront parfaitement, mais cette méthode est un peu moins robuste (la souris réapparaît parfois sans raison).

Voir aussi: Problèmes

1.30 Touches d'appel de Yak

En cliquant sur le gadget "Touches d'appel..." , une nouvelle fenêtre s'ouvre pour vous permettre de créer, éditer et effacer des touches d'appel. Transformer une touche du clavier en une touche d'appel signifie lorsque cette touche est enfoncée, Yak exécute une action (que vous pouvez choisir parmi les nombreuses disponibles). Les touches d'appel sont définies par une chaîne de caractères, ce qui est une méthode très souple pour définir des événements. Pour plus de détails sur la façon de définir des touches d'appel, voir Détails sur les touches d'appel.

Vous pouvez avoir autant de touches d'appel que vous le souhaitez, et chaque action peut être rattachée à plus d'une touche d'appel.

Il y a 2 listes dans la fenêtre de définition des touches d'appel; celle de gauche énumère toutes les actions possibles tandis que celle de droite énumère la liste des hotkeys actuellement rattachées à l'action sélectionnée.

Pour ajouter une nouvelle hot key, sélectionnez d'abord l'action que vous souhaitez exécuter (en cliquant sur son nom dans la liste de gauche). Puis cliquez sur le gadget "Ajouter" sous la liste des touches d'appel. Le string gadget sous cette liste sera alors activé et vous pourrez taper la chaîne de caractères décrivant la touche d'appel.

Pour certaines actions, d'autres gadgets seront activés. Le gadget "Options" (un cycle gadget sous la liste des Actions) détermine ce qu'il arrive aux écrans quand la touche d'appel est enfoncée. Ces options sont ne rien faire (No Screen Change), amener l'écran du Workbench à l'avant-plan (Workbench to Front), et amener à l'avant-plan l'écran public par défaut (Default PubScr to Front). Typiquement, ceci sera utilisé pour les touches d'appel "Commande Dos" pour montrer automatiquement l'écran sur lequel la fenêtre s'ouvre.

Le gadget Paramètre est un string gadget qui est activé pour certains types d'actions et qui fournit un moyen de définir une chaîne de caractères devant être rattachée à une touche d'appel.

Vous pouvez éditer une touche d'appel existante en la sélectionnant et en éditant les différents gadgets nécessaires. Et bien entendu, le gadget "Effacer" efface une touche d'appel de la liste de Yak.

Quand vous avez fini d'éditer vos touches d'appel, cliquez sur le gadget "Revenir..." pour revenir à la fenêtre principale de configuration, ou le gadget de fermeture pour "cacher" Yak. Souvenez vous qu'il vous faut sauvegarder vos changements en sélectionnant l'item "Sauver les Préfs" dans le menu de la fenêtre principale de configuration.

Voir Actions des touches d'appel

1.31 Actions des Hotkeys

Liste des actions disponibles :

- Commande Dos
- Fermer une fenêtre
- Zipper une fenêtre
- Minimiser une fenêtre
- Maximiser une fenêtre
- Cycler les fenêtres
- Ouvrir une palette
- Ecran en avant
- Ecran en arrière
- Activer le Workbench
- Centrer un écran
- Eteindre l'écran
- Insérer du texte
- Insérer la date
- Configurer Yak

1.32 Command DOS

Paramètre: commande à exécuter

Exécute la commande Dos spécifiée en paramètre. Notez que vous pouvez utiliser le gadget d'options pour amener à l'avant plan l'écran du workbench ou l'écran public par défaut.

La commande est exécutée de manière asynchrone, si bien qu'il n'est pas nécessaire de préfixer la commande Dos avec la commande run. Si la commande ainsi lancée nécessite un affichage ou bien une entrée au clavier, une fenêtre console sera ouverte. Vous pouvez bien sûr spécifier des redirections (comme dans un shell).

Exemples:

Une touche d'appel pour ouvrir un shell:

Cette commande est traditionnellement attachée à la hot key "lcommand esc", la mienne est configurée ainsi :

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

Notez l'usage du flag ALT dans la spécification de la console, qui est plutôt mal documenté (lire "non mentionné"). En fait, j'utilise 2 touches d'appel, une pour lancer un shell normal et l'autre pour lancer un CShell.

Une touche d'appel pour libérer la mémoire inutilisée:

SAS/C utilise des bibliothèques partagées qui peuvent souvent "gaspiller" de la très précieuse mémoire graphique (CHIP). Je me suis configuré une touche d'appel appelant la commande "avail >nil: flush" qui libère cette mémoire.

Une touche d'appel pour lister le contenu de chaque nouvelle disquette insérée dans le lecteur interne:

Choisir la touche d'appel "diskinserted", et prendre comme paramètre "Dir df0:".

Voir aussi: [Problems](#) [Détails sur les touches d'appel](#)

1.33 Fermer la fenêtre

Paramètre: AUCUN

Ferme la fenêtre actuellement active (c'est équivalent à cliquer le gadget de fermeture de la fenêtre).

1.34 Zipper la fenêtre

Paramètre: AUCUN

Zippe la fenêtre actuellement active (c'est équivalent à cliquer le gadget de "Toggle size" de la fenêtre).

1.35 Minimiser la fenêtre

Paramètre: AUCUN

Minimise la fenêtre actuellement active en la rendant aussi petite que possible.

1.36 Maximiser la fenêtre

Paramètre: AUCUN

Maximise la fenêtre actuellement active en la rendant aussi grande que possible.

1.37 Cycler les fenêtres

Paramètre: AUCUN

Amène à l'avant plan la fenêtre située le plus à l'arrière plan. Utile pour accéder aux fenêtres complètement occultées par d'autres.

1.38 Ouvrir une palette

Paramètre: AUCUN

Ouvre une palette sur l'écran à l'avant-plan. La palette est asynchrone et vous pouvez en ouvrir autant que vous le souhaitez (dans la limite de la mémoire disponible). Toutefois, tant qu'une palette est ouverte, vous ne pouvez pas fermer Yak. Le gadget d'options est valide pour cette touche d'appel si bien que vous pouvez si vous le souhaitez ouvrir une palette spécifiquement sur l'écran du Workbench.

NOTE: Vous devez avoir installé la `reqtools.library` sur votre système pour pouvoir utiliser cette action.

ATTENTION: Vous devez fermer la palette avant de fermer l'écran auquel elle est rattachée, autrement vous aurez dans le meilleur des cas un écran impossible à fermer et dans le pire un crash système peut se produire.

1.39 Ecran en avant

Paramètre: AUCUN

Met à l'avant-plan, l'écran le plus à l'arrière-plan.

Voir aussi: ScreenActivate

1.40 Ecran en arrière

Paramètre: AUCUN

Met à l'arrière plan, l'écran à l'avant plan.

Voir aussi: ScreenActivate

1.41 Activer le Workbench

Paramètre: AUCUN

Active une fenêtre du Workbench (et si nécessaire, amène l'écran du Workbench à l'avant-plan). Ceci vous permet d'accéder facilement aux menus du Workbench sans avoir besoin de chercher une fenêtre à activer (si, par exemple, vous avez une fenêtre shell occupant tout l'écran).

1.42 Centrer l'écran

Paramètre: AUCUN

Centre horizontalement l'écran à l'avant plan.

1.43 Eteindre l'écran

Paramètre: AUCUN

Eteint immédiatement l'affichage.

Voir aussi: ScreenTimeOut .

1.44 Insérer du texte

Paramètre: text to be inserted

Insère le texte spécifié en paramètre dans le flot de données en entrée. Cette chaîne de caractères peut comporter des caractères spéciaux qui sont interprétés par Yak :

```
\n    converti en retour à la ligne
\r    converti en retour à la ligne
\t    converti en tabulation
\     converti en backslash \
<hotkey desc> converti en la touche d'appel spécifiée
\<    converti en <
```

Grâce à ce pré-traitement, les chaînes à insérer peuvent rendre de grands services. Par exemple, j'ai configuré une touche d'appel pour insérer mon nom et la date de cette manière :

```
"Martin W Scott, <lcommand d>"
```

Ici, la touche d'appel "lcommand d" est une autre touche d'appel de Yak que j'ai configurée pour insérer la date. En utilisant des chaînes plus compliquées, vous pouvez créer des macros simples pour d'autres programmes.

INCONVENIENT: Les chaînes de touche d'appel imbriquées (ex: la date ci-dessus), bien que très utiles, doivent être employées avec précaution. En particulier, il faut absolument éviter des définitions récursives telles que:

```
f1 insert text "<f2>"
f2 insert text "<f1>"
```

Dans ce cas, l'appui sur f1 ou f2 crée une boucle infinie. Si malgré tout, il vous arrivait de faire une telle chose, lancez le programme exchange situé dans le répertoire Commodities du répertoire Tools de votre système et rendez Yak inactif. Puis, sélectionnez Montrer et effacez/redéfinissez les touches d'appel fautives.

Une autre chose à savoir est que les chaînes qui appellent d'autres touches d'appel (ex: l'insertion de la date ci-dessus) peuvent ne pas fonctionner comme espéré. Supposons que la chaîne en paramètre soit "<lcommand d>\n". Ceci en fait insérerait un saut de ligne et ENSUITE la date, parce qu'au moment où Yak traite la touche d'appel "lcommand d" touche d'appel, le saut de ligne a déjà été transmis à la fenêtre active.

1.45 Insérer la date

Paramètre: chaîne de format de date

Insère la date dans le flot de données entrant (et donc dans la fenêtre active window). Si vous utilisez le système 2.1 ou plus, vous pouvez personnaliser le format d'affichage de cette date selon vos goûts. Ce format est spécifié dans la chaîne en paramètre. Voir Spécifier un format de date DateFmt}. Si pour votre malheur, vous êtes encore en 2.0, la date sera affichée dans le format standard JJ-MMM-AA.

Exemple: le format "%e %B %Y" génère des dates de la forme "1 Mai 1993".

1.46 Configurer Yak

Paramètre: AUCUN

Ouvre la fenêtre de configuration de Yak (si nécessaire en amenant d'abord à l'avant-plan l'écran du Workbench puis en ouvrant la fenêtre).
C'est la même fonction qu'effectue la touche d'appel CX_POPKEY.

1.47 Motifs AmigaDos

Les Motifs (Patterns) AmigaDos sont utilisées pour inclure/exclure une liste nominative d'écrans/fenêtres d'un certain traitement. Ces patterns de spécification aident à une meilleure compatibilité avec d'autres programmes que vous utilisez.

La correspondance à une pattern est dépendante des majuscules/minuscules. "Amiga" n'est pas pareil que "AMIGA". Les patterns AmigaDos standards disponibles sont :

| | |
|---------|---|
| ? | Correspond à n'importe quel caractère unique. |
| # | Correspond à l'expression suivante 0 ou plusieurs fois. |
| (ab cd) | Correspond à l'un ou l'autre des éléments séparés par ' '. ~ |
| | Négation de l'expression suivante. Cela correspond à toutes les chaînes qui ne correspondent pas à l'expression (par exemple ~(truc) correspond à toutes les chaînes qui ne sont pas exactement "truc". |
| [abc] | Classe de caractères: correspond à l'un des caractères de la classe. |
| a-z | Intervalles de caractères (seulement dans les classes de caractères). |
| % | Correspond à toujours à 0 caractères (utile dans "(foo bar %)") |
| * | Synonyme de "#?", non disponible par défaut. Uniquement disponible si l'option Wildstar est mise. |

Si vous n'êtes pas familiarisés avec les patterns, vous trouverez sûrement cela plutôt décourageant. Consultez votre manuel amigados pour de plus amples détails. Voici 2 choses élémentaires que vous pouvez désirer: soit une liste finie de noms pour laquelle la fonction doit être applicable, ou une liste finie de noms pour laquelle elle ne doit pas l'être. Pour rendre applicable une fonctionnalité à un nombre N fini d'objets nommés de "nom1" à "nomN", utilisez :

```
(nom1|nom2| ... |nomN)
```

Et pour rendre inapplicable une fonctionnalité à ces noms, ajoutez un tilde ~ devant :

```
~(nom1|nom2| ... |nomN)
```

Un exemple: Je ne veux pas qu'AutoPopToFront poppe la fenêtre du Workbench ou

de Protex (le traitement de textes d'Arnor), donc je les exclue grace à la pattern suivante :

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

Notez que le second 'nom' is en fait une pattern, à laquelle correspond tout titre contenant le texte "Arnor".

Un autre exemple: Je ne veux pas d'autoactivation sur l'écran de Directory Opus. Comme il ne montre pas le titre de son écran, j'ai dû utiliser ARTM (ou Xoper) pour trouver le nom de l'écran qui était "DOPUS.1". En fait si on lance deux fois DOpus, le "1" devient un "2", j'ai donc décidé d'exclure tous les écrans de DOpus de cette manière :

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

Note 1: Spécifier le titre de l'écran du Workbench est plutôt difficile car il n'arrête pas de changer, selon la fenêtre active. Dans la plupart des cas, une pattern telle que #?Workbench#? fonctionnera, mais quelques applications utilisent le titre de l'écran du workbench pour afficher une description de leur application.

Note 2: Les écrans et les fenêtres dont les titre ne sont pas positionnés (égaux à NULL) ne peuvent pas être exclus par une pattern.

1.48 Spécifier un format de date

Pour la fonction d'insertion de date, vous devez utiliser une chaine de format de date du type locale.library (et donc vous avez besoin de l'AmigaDos 2.1). Les options de formatage disponibles dans la locale.library sont les suivantes

```
%a - jour de la semaine en abrégé
%A - jour de la semaine
%b - nom du mois en abrégé
%B - nom du mois
%c - équivalent à "%a %b %d %H:%M:%S %Y"
%C - équivalent à "%a %b %e %T %Z %Y"
%d - jour en chiffres précédé d'un zéro si nécessaire
%D - équivalent à "%m/%d/%y"
%e - jour en chiffres précédé d'un espace
%h - nom du mois en abrégé
%H - heure (sur 24 heures) précédée d'un zéro si nécessaire
%I - heure (sur 12 heures) précédée d'un zéro si nécessaire
%j - date selon le calendrier Julien
%m - mois en chiffres précédé d'un zéro si nécessaire
%M - minutes précédées d'un zéro si nécessaire
%n - insère un saut de ligne
%p - chaines AM ou PM (notation anglo-saxone pour l'heure)
%q - heure (sur 24 heures)
%Q - heure (sur 12 heures)
%r - équivalent à as "%I:%M:%S %p"
%R - équivalent à "%H:%M"
```

%S - secondes précédées d'un zéro si nécessaire
%t - insère une tabulation
%T - équivalent à "%H:%M:%S"
%U - numéro de semaine en prenant Dimanche comme premier jour de la semaine
%w - numéro du jour de la semaine
%W - numéro de semaine en prenant Lundi comme premier jour de la semaine
%x - équivalent à "%m/%d/%y"
%X - équivalent à "%H:%M:%S"
%y - année sur 2 chiffres précédée d'un zéro si nécessaire
%Y - année sur 4 chiffres précédée de zéros si nécessaire

Cette liste est plutôt exhaustive et devrait couvrir tous les besoins; vous pouvez aussi insérer votre propre texte librement dans la chaîne de format. Quelques exemples:

| | | |
|---------------------|-----------------|------------------------|
| "Il est %X" | donne (par ex.) | "Il est 20:44:16" |
| "Passez un bon %A!" | donne (par ex.) | "Passez un bon Lundi!" |

Pour plus de détails, consultez les AutoDocs de la locale.library si vous les avez.

1.49 Problèmes

Il y a quelques problèmes dont je suis au courant. Premièrement, un shell créé par la touche d'appel PopCommand ne possède ni la stack (la taille de la pile) ni le répertoire courant définis au moment du boot (dans la Startup-Sequence). Par contre, il conserve le path. Votre fichier Shell-Startup doit définir le répertoire courant et la stack dont vous avez besoin.

La touche d'appel "Fermer la fenêtre" ne marche pas pour les fenêtres shell, c'est parce qu'elles gèrent différemment les événements Intuition.

Si la fonction d'extinction de la souris de Yak ne vous satisfait pas, vous pouvez essayer l'utilitaire MouseBlank de Commodore (WB3.0) qui devrait faire disparaître la souris sur tous les types d'affichages correctement.

NOTE AUX UTILISATEURS D'AMOS: Je hais AMOS (mais ce n'est que mon opinion), en partie parce qu'il est hostile envers le système. Il accapare complètement le flot d'entrée, si bien que les mouse blankers (comme celui de Yak) échouent parce qu'ils pensent qu'il n'y a aucune entrée et ainsi la souris ne réapparaît pas parce qu'aucun mouvement de souris n'est détecté. Du fait que Yak utilise une méthode plutôt mauvaise pour effacer la souris, des problèmes peuvent survenir (la souris disparaît et ne réapparaît pas). Voici 2 solutions:

- 1) Utilisez la méthode 'Copper'
- 2) Mettre le délais d'extinction de la souris à zéro. La souris disparaîtra toujours dès que vous taperez une touche du clavier, mais pas en fonction du temps d'inactivité de la souris.

Voilà, le problème d'AMOS et du pointeur qui disparaît devrait être résolu.

1.50 Historique du programme

(* = nouvelle fonction)

v1.52 * Yak est maintenant localisé et possède un catalogue français.

* L'installation se fait maintenant par un script Installer.

- Nouvelle numérotation des versions pour utiliser RCS.

- Petit bug corrigé dans les NewMenus en 3.0.

- Petits changements dans la documentation et l'interface.

- Maintenant compilé avec le SAS C 6.3

v1.5a - La taille de la pile nécessaire a été augmentée.

- Le type d'outil NOICON a été remplacé par APPICON.

- Amélioration du nouveau système de touches d'appel.

v1.5 * Système extensible de touches d'appel ajouté.

* Refonte de l'interface graphique utilisateur.

* La documentation est maintenant au format AmigaGuide (non, vraiment?)

* L'Insertion de la date fonctionne maintenant aussi sur les machines en 2.0.

- Nouveau format de fichier de préférences, et maintenant utilise aussi un fichier pour les touches d'appel.

Les fichiers de préférences des versions précédentes ne sont plus reconnus.

v1.4a * Ajout du sélecteur 'RMB Activate' - la fenêtre sous la souris est activée quand le bouton droit de la souris est pressé.

* Nouvelle méthode améliorée de mouse-blanking: la Méthode 'Sprites' est la même qu'avant, mais la méthode 'Copper' a été complètement réécrite; elle ne fait toujours disparaître que le sprite 0 mais est maintenant compatible avec LacePointer.

- Maintenant compilé avec le SAS/C 6.2.

v1.4 - Vérification maintenant effectuée sur les intervalles des gadgets numériques

- NewLookMenus pour le WB 3.0.

- v1.3f * La fenêtre de configuration s'adapte maintenant à la fonte utilisée par l'écran.
- * La fonction ScreenToBack en cliquant sur la souris maintenant active la première fenêtre sur l'écran à l'avant plan quand il n'y a pas de fenêtre sous la souris (cela aide des programmes comme MagicMenu).
 - * L'écomiseur d'écran (Screen blanker) ouvre maintenant un écran dans le même mode d'affichage que l'écran à l'avant-plan.
 - Encore un bug de moins dans le mouse-blanking.
 - Conflits entre la touche d'appel window-cycle et la fonction keyactivation résolus, malgré tout elles ne sont pas vraiment faites pour fonctionner ensemble.
 - Le cyclage des fenêtres (par touche d'appel) fait maintenant aussi cycliser les fenêtres du Workbench; à ses tout débuts, Yak émulait la fonction de IHelp (CBM OS2.0) de cyclage des fenêtres d'application seulement. Mais on m'a tellement souvent dit que cette fonction ne marchait pas que je me suis dit que les gens voulaient réellement cycliser TOUTES les fenêtres.
 - Encore quelques enforcer-hits enlevés (oh comme j'aimerais avoir un 030 etc., ainsi je pourrais corriger tout ça moi même, au lieu de compter sur les autres pour m'en informer).
- v1.3e - Chaîne de version ajoutée, visible en tapant "c:version Yak"
- Vous avez maintenant le choix de la méthode de mouse-blanking (ainsi, finies les remarques incendiaires...)
- v1.3d - Correction d'un bug dans la routine de KeyActivate (elle perdait le premier caractère tapé).
- v1.3c - Correction d'un petit enforcer hit qui faisait parfois échouer le cyclage de fenêtres.
- v1.3b * Nouvelle pattern ClickWindows, permet de spécifier sur quelle fenêtre la fonction click-to-front est applicable.
(n.b. ça n'a aucun effet sur click-to-back).
- * La durée au bout de laquelle le pointeur de souris disparaît est maintenant configurable.
 - Le volume du click est maintenant spécifié par un gadget numérique.
- v1.3a - Maintenant tous les appels à Intuition se font à l'extérieur de l'input-handler pour éviter des blocages possibles (Merci à Eddy Carroll pour l'avoir signalé).
- * Nouvelle touche d'appel: Screen to back. Avant, ceci était non configurable et fixé à LCommand m.
 - * Nouvelle touche d'appel: Blank display. Eteint l'écran immédiatement.
-

- * On peut maintenant envoyer un écran à l'arrière plan en cliquant dans sa barre de titre (cela b'était pas permis avant pour éviter des conflits avec le système).
 - * La palette fonctionne maintenant en mode asynchrone, et vous pouvez en ouvrir autant que vous voulez à la fois. Elles doivent néanmoins être fermées avant que leurs écrans respectifs ne soient fermés, et avant que vous ne stoppiez Yak.
 - Maintenant l'insertion d'une disquette arrête l'extinction d'écran.
 - Le centrage des écrans par touche d'appel a été ajusté d'un pixel, pour être en accord avec Display, AutoCentre etc. (J'avais tort...)
 - Quand Yak est démarré depuis le workbench, il prend SYS: comme répertoire courant; celui-ci est ensuite hérité par les shells créés par PopCommand.
 - Encore quelques problèmes de mouse-blanking corrigés. Aussi, un bug du screen-blanking corrigé qui faisait clignoter le pointeur de souris - comme pratiquement personne n'utilise le blanker de Yak, cela n'avait pas été remarqué jusqu'à présent.
 - Petit bug corrigé dans la priorité d'allocation de canal.
 - Un bug corrigé dans l'annulation d'une touche d'appel (en effaçant le string gadget). Auparavant, lorsqu'on annulait une touche d'appel pour la cession en cours, l'ancienne définition était toujours affichée. Comment ai-je pu laisser passer ça ?
- v1.3
- * L'exécutable utilise maintenant des overlays. Il utilise maintenant moins de mémoire que les précédentes versions, mêmes avec les nouvelles fonctions!
 - * Possibilité de spécifier sur quels écrans utiliser AutoActivate et ClickToFront/Back à l'aide de patterns AmigaDos.
 - * L'insertion de la date utilise maintenant des chaînes de format de type locale.library. (Malheureusement, l'insertion de la date nécessite maintenant le WB 2.1).
 - * AutoPopToFront a été ajouté.
 - * Le mouse-blanking a été amélioré: maintenant Yak ne fait disparaître que le sprite 0, si bien que les programmes qui utilisent d'autres sprites ne sont pas affectés.
 - * PopCommand maintenant conserve les chemins (paths) mais pas le répertoire courant ni la taille de la pile (stack) définies dans la startup-sequence.
 - L'Auto-Activation et la key-activation fonctionnent avec plus de discernement.
 - Le cyclage des fenêtres (par touche d'appel) fonctionne maintenant sur tous les écrans (avant cela ne fonctionnait que sur le WB, mais je ne
-

l'avais pas mentionné dans les docs).

- NB - Le code des tests permettant d'éviter des conflits était plutôt sévère en 1.2c dans certains cas, et inutile après avoir changé le fonctionnement de l'AutoActive de Yak. Maintenant, Yak n'AutoActive (et n'AutoPop) que lorsqu'aucun qualifieur ou bouton de la souris n'est pressé. La vérification des conflits possibles a donc été enlevée. Il ne devrait pas y avoir de problèmes avec les programmes se comportant correctement.
- NB - Le fichier de configuration a changé; il s'appelle maintenant S:Yak.prefs au lieu de S:Yak.prefs; c'est parce que le format de ce fichier a changé. Un programme de conversion nommé 'Convert' est fourni pour créer un fichier 1.3 à partir d'un fichier 1.2.
- [La présentation de la fenêtre de préférences a changé; elle est plus jolie maintenant (selon moi).]
- v1.2c - Inclusion de la documentation sur les Hot Keys (adaptée de la documentation de Stefan Becker pour ToolManager)
- Un "enforcer hit" de moins. (mais il y en a peut être encore)
- Amélioration du test pour éviter les conflits; maintenant la fonction AutoActive de Yak fonctionne maintenant avec Dragit V2. par exemple.
- v1.2b - la longueur maximale des définitions des hot keys est passée de 29 à 49 pour permettre (presque) toutes les définitions possibles.
- v1.2a - léger bug dans v1.2; n'effaçait pas l'ancienne hot key après que définition existante ait été changée.
- v1.2
- * le pointeur de souris ne disparaît plus quand la touche Amiga de droite est enfoncée, si bien que le pilotage de la souris à partir du clavier est utilisable.
 - * active les écrans (= fenêtres sur les écrans) quand elles sont manipulées, similaire à WindX.
 - * maintenant rajoute optionnellement une AppIcon sur l'écran du Workbench quand elle est double-cliquée, la fenêtre des préférences apparaît.
 - entrer une chaîne vide comme touche d'appel (pour l'effacer) créerait une touche d'appel attachée à la touche '. Je considère ceci comme un bug de commodities.library. Maintenant une chaîne vide effacera la touche d'appel.
 - Méchant bug dans auto-activer (clavier ou souris) corrigé. Avant, Yak choisissait parfois la mauvaise fenêtre (sur le mauvais écran) si plus d'un écran était visible.
 - Change maintenant de fenêtre active même quand la fenêtre actuelle a
-

un requester en cours (mais si la fenetre active a un string-gadget actif, il ne sera pas désactivé). Cela semble plus utile (pour moi en tout cas).

- changé l'option 'Screen Cycle' en 'Screen To Back' pour éviter une confusion avec la touche d'appel 'Screen Cycle' (suis-je bête ...)

v1.1 * utilise maintenant un fichier de configuration (s:Yak.config) au lieu de 10 millions de tooltypes. Seuls les tooltypes standards sont maintenant reconnus.

- * ajouté la possibilité de faire cycler les fenêtres et les écrans depuis le clavier (comme IHelp).

- * ajouté l'option screen-center touche d'appel.

- * ajouté l'option wildstar.

- [légerement changé la présentation de la fenêtre de configuration, menu ajouté]

- Bug corrigé dans la fonction click-to-back (conflit avec PowerSnap); Elle opère maintenant uniquement si aucun "qualifier" n'est pressée.

- Bug corrigé dans le blanking de la souris, le pointeur de la souris réapparaissait périodiquement.

- les touches d'appel produisent maintenant un "clic" si keyclick est configuré.

v1.0 - première version.

1.51 Compiler Yak

Yak a été écrit pour être compilé sous SAS/C, and à partir de la version 1.4a, en utilisant le SAS/C version 6.2. Le code se compile proprement avec l'option de vérification ANSI, si bien que les utilisateurs d'autres compilateurs C ANSI devraient avoir peu de problèmes à recompiler Yak - les seuls endroits pouvant nécessiter des adaptations sont ceux où sont utilisés des mots clés spécifiques au SAS/C (comme par exemple __saveds).

Yak utilise des overlays, mais il peut être compilé sans ceux-ci en enlevant simplement la définition USE_OVERLAYS dans SCOPTIONS.

GTB 2.0b a été utilisé pour générer l'interface utilisateur. Le source généré (popup.c, popup.h) ne nécessite que des modifications minimales pour utiliser les menus NewLook sous WB3.0 et plus; ces changements sont indiqués par des commentaires commençant par 4 astérisques, par exemple /**** ADDED */.

1.52 Remerciements

Yak est entièrement écrit en C, et compilé avec le SAS/C 6.3. Merci à Reza Elghazi pour son aide dans la conversion de 5.10a vers 6.2.

La fenêtre de configuration a été créée en utilisant l'excellent (à part le petit problème de la version 2.0 :) GadToolsBox 2.0b, de Jaba Development.

Yak utilise reqtools.library, qui est Copyright Nico François. Mes remerciements vont aussi à Mike Meyer et. al. pour Mg3, Steve Koren pour SKsh, Matt Dillon pour DMouse (qui répondit à beaucoup de mes questions), et Kai Iske pour KCommodity d'où vient le son du clic clavier. Yak utilise aussi WB2CLI, un petit module a linker tres utile écrit par Mike Sinz. Merci à Pierre Carette et Sylvain Rougier pour BrowserII, à Heinz Wrodel pour son excellent portage de RCS 5.6 sur Amiga et Martin Korndörfer pour MagicMenu.

La documentation sur les Hot Keys est tirée de la distribution de ToolManager, avec l'aimable permission de Stefan Becker.

Merci à Stefan Sticht pour son utilitaire MouseBlanker du domaine public d'où vient la méthode 'Copper' de mouse-blanking.

Merci à tous les gens qui m'ont écrit à propos de Yak, pour leurs suggestions et les bugs trouvés.

Merci à mes copains de la mailing liste amiga francophone qui m'ont bien aidé pour la localisation française.

Et finalement, un grand merci à Martin Scott pour avoir créé Yak.

1.53 Contacter l'auteur

Envoyez vos rapports de bug, commentaires, suggestions, donations, cartes postales etc... à:

Gaël MARZIOU
Cidex 103
38920 CROLLES
FRANCE

ou bien par e-mail (de préférence) : gael@gnlab030.grenoble.hp.com

1.54 Détails sur les touches d'appel

[Extrait de TM_English.doc faisant partie de ToolManager]

Comment définir une Hot Key

Ce chapitre decrit comment définir une Hot Key (une touche d'appel) dans une

chaîne de description, qui est ensuite analysée par Commodities. Chaque fois qu'une touche d'appel est activée Commodities génère un événement qui est utilisé par Yak (ou une autre commodity active). Voici la syntaxe utilisée pour la chaîne de description d'une touche d'appel :

```
[<class>] {[-][<qualifier>]} [-][upstroke] [<key code>]
```

Tous les mots clés peuvent être écrits indifféremment en majuscules ou minuscules.

'class' décrit la classe de l'événement (InputEvent). Ce paramètre est optionnel et en son absence c'est la classe 'rawkey' qui est utilisée par défaut. Voir les classes d'événements "InputEvent".

Les "qualifieurs" sont des signaux qui doivent être positionnés ou effacés au moment où la touche d'appel est activée; autrement aucun événement ne sera généré. Pour chaque qualifier qui doit être positionné, vous devez donner son mot clé. Tous les autres qualifieurs sont ignorés par défaut. Si vous voulez ignorer un qualifier, ajoutez juste un '-' devant son mot clé. Voir la liste des qualifieurs.

Normalement un événement touche d'appel est généré quand une touche est enfoncée. Si vous voulez que l'événement soit généré quand la touche est relâchée, ajoutez le mot clé 'upstroke'. Quand vous souhaitez générer un événement sur une touche enfoncée ou relâchée, utilisez '-upstroke'.

Le "key code" (le code d'une touche du clavier) dépend de la classe d'événement choisie. Voir Codes des touches.

Note: Choisissez vos touches d'appel avec soin, car Commodities a une priorité haute dans la chaîne du gestionnaire d'événements. (c.a.d que vous pourriez prendre le dessus sur une définition existante).

Classes d'événements (InputEvent)

=====

Commodities supporte la plupart des classes d'événements générés par l'input.device. Cette section décrit les classes les plus utiles pour les Touches d'appel de Yak.

'rawkey'

C'est la classe par défaut et elle couvre tous les événements clavier. Vous devez spécifier le code d'une touche pour cette classe. Voir aussi la section "rawkey key codes".

'rawmouse'

Cette classe décrit tous les événements relatifs aux boutons de la souris. Vous devez spécifier le code d'une touche pour cette classe. Voir aussi la section "rawmouse key codes".

'diskinserted'

Les événements de cette classe sont générés quand une disquette est insérée dans un lecteur. Cette classe n'a pas de "key codes".

'diskremoved'

Les événements de cette classe sont générés quand une disquette est

retirée d'un lecteur. Cette classe n'a pas de "key codes".

Qualifiers

=====

Certains synonymes de mots clés ont été ajoutés à la Commodities V38., ces derniers sont repérés par une '*'.

'lshift', 'left_shift' *
touche shift gauche.

'rshift', 'right_shift' *
touche shift droite.

'shift'
L'une ou l'autre des touches shift.

'capslock', 'caps_lock' *
La touche Caps lock.

'caps'
L'une ou l'autre des touches shift et de la touche Caps lock.

'control', 'ctrl' *
La touche Ctrl (Control).

'lalt', 'left_alt' *
La touche Alt de gauche.

'ralt', 'right_alt' *
La touche Alt de droite.

'alt'
L'une ou l'autre des touches Alt.

'lcommand', 'lamiga' *, 'left_amiga' *, 'left_command' *
La touche Amiga gauche.

'rcommand', 'ramiga' *, 'right_amiga' *, 'right_command' *
La touche Amiga droite.

'numericpad', 'numpad' *, 'num_pad' *, 'numeric_pad' *
Ce mot clé doit absolument être utilisé pour toute touche située sur le pavé numérique.

'leftbutton', 'lbutton' *, 'left_button' *
Bouton gauche de la souris. Voir la note ci-dessous.

'midbutton', 'mbutton' *, 'middlebutton' *, 'middle_button' *
Bouton du milieu de la souris. Voir la note ci-dessous.

'rbutton', 'rightbutton' *, 'right_button' *
Bouton droit de la souris. Voir la note ci-dessous.

'repeat'
Ce qualifier est utilisé quand la fonction "répétition" du clavier est active. Utile uniquement pour la classe 'rawkey'.

Note: Commodities V37 a un bug qui empêche l'utilisation de 'leftbutton', 'midbutton' et 'rbutton' en tant que qualifieurs. Ce bug est corrigé dans la version V38.

Codes des touches

=====

Chaque classe d'événement a ses propres codes de touches:

classe 'rawkey'

Certains synonymes de mots clés ont été ajoutés à la Commodities V38., ces derniers sont repérés par une '*'.

'a'-'z', '0'-'9', ...
caractères ASCII.

'f1', 'f2', ..., 'f10', 'f11' *, 'f12' *
touches de fonction.

'up', 'cursor_up' *, 'down', 'cursor_down' *
'left', 'cursor_left' *, 'right', 'cursor_right' *
touches des flèches du curseur.

'esc', 'escape' *, 'backspace', 'del', 'help'
'tab', 'comma', 'return', 'space', 'spacebar' *
touches spéciales.

'enter', 'insert' *, 'delete' *
'page_up' *, 'page_down' *, 'home' *, 'end' *
touches du pavé numérique. Chacun de ces codes doit être utilisé avec
le qualifieur 'numericpad' !

classe 'rawmouse'

Ces mots clés ont été ajoutés à la Commodities V38. Ils ne sont donc pas disponibles dans la V37.

'mouse_leftpress'
Presser le bouton gauche de la souris.

'mouse_middlepress'
Presser le bouton du milieu de la souris.

'mouse_rightpress'
Presser le bouton droit de la souris.

Note: Pour utiliser un de ces codes, vous devez aussi mettre le mot clé du qualifieur correspondant, par ex :

rawmouse leftbutton mouse_leftpress

Exemples de touches d'appel

=====

``ralt t'`

Maintenez enfoncée la touche Alt droite et appuyez sur "t"

``ralt lalt t'`

Maintenez enfoncées les touches Alt droite et gauche et appuyez sur "t"

``alt t'`

Maintenez enfoncée l'une ou l'autre des touches Alt et appuyez sur "t"

``rcommand f2'`

Maintenez enfoncée la touche Amiga droite et appuyez sur la deuxième touche de fonction.

``numericpad enter'`

Appuyez sur la touche "enter" du pavé numérique.

``rawmouse midbutton leftbutton mouse_leftpress'`

Maintenez enfoncé le bouton du milieu de la souris et appuyez sur le bouton gauche de la souris.

``diskinserted'`

Insérez une disquette dans un lecteur.
