ilbm

Michael Zucchi

COLLABORATORS			
	TITLE : ilbm		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Michael Zucchi	July 22, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

ilbm

1.1 ilbm.guide

```
Module de chargement d'images IFF ILBM pour AmigaE2.5+

© 1993 Michael Zucchi
Tous droits réservés

Ce document décrit le module ilbm.m, qui inclu des commandes de chargement
et d'affichage (si possible) d'image IFF. Les fonctions données sont faites
pour être utilisé le plus simplement possible, donnant beaucoup de flexibilité.

Les ections suivantes sont disponibles:

        Survol          quelques idées derrière le module

       Fonction du Module

            ilbm_New()          pour ouvrir une image
            ilbm_Dispose()          pour nettoyer

            ilbm_PictureInfo()  pour prendre la taille et palette d'un image etc
            ilbm_LoadPicture()  charge les données dans l'écran et bitmaps

            ilbm_FreeBitMap()    pour libérer un bitmap alloué par LoadPicture

        Exemples

NOTE: Ce module a besoin du Workbench 2.0 (V36) ou plus!  SVP
assurez vous que cette version des bibliothèques système est présente avant
d'utiliser ces fonctions.

NOTE!!!  A cause d'u oubli, uniquement des ilbm COMPRESSÉS marche.
Ce sera bientôt corrigé... J'espère ? (des ilbm non compressés
ne sont pas commun de toute façon)
```

1.2 Survol du Module

Pas grand chose à dire en fait - ce module est juste fait pour charger/sauver des IFFs!
Fait pour être utilisé simplement pour charger des images ilbm sur écrans, ou dans des bitmaps pour être 'blitter' plus tard, ou n'importe où d'autre où un ilbm peut être utile.

Une chose - ce module marche sur les systèmes V36, malgré tout, sur les systèmes V39. Les nouvelles fonctions de la graphic.library sont utilisés le plus possible (LoadRBG32() pour les palettes 34 bits sur les machines AGA+.

Le futur

Comme annoncé, le module est idéal pour charger des images pour affichage. Une autre idée qui peut être implémenté est une fonction de chargement pour le mode 'chunky' (eg ILBM_CHUNKY) où les données sont convertis au format octet-par-pixel avant de sortir en un tableau d'octet.

Une fonction de sauvergarde serait aussi utile - je ne l'ai pas encore implémenté par manque de temps, et aussi pour garder un petit module.

En fait ...

Ce type de chose devrait être mieux gérer avec les datatypes. Malheureusement ils sont vraiment peu pratique a utiliser - et pas très bon. Je voyais ce module au début comme un chargeur d'images ilbm pour des jeux/applications, plutôt que comme afficheur d'images.

1.3 Le gars qui l'a écrit

Le décompresseur iff, que j'ai écrit il y a quelques temps pour zgif, était raisonnablement rapide, mais il n'était pas fait uniquement pour sa vitesse!

En ce moment, j'étudie 'de temps en temps (:-) en vue d'obtenir le diplome d'ingénieur sur système informatique (Computer Systems Engineering) à l'Univerity d'Australie du Sud. (1994=final year)
Je suis aussi 'Zed' de chez FRONTIER à mes heures anti-os.

Je peux être contacté aux adresses suivantes:

Internet email:
 9107047w@lux.levels.unisa.edu.au
       jusqu'à la fin '94

'Real Mode' (tm) mail:
   Michael Zucchi
   PO BOX 824
   Waikerie
   South Australia 5330
       Lent mais très sûr - jusqu'à ce que ma mère vende la maison :)

   Michael Zucchi
   110 Dunrobin Rd
   Warradale
   South Australia 5046

à ma porte - jusqu'à ce que je déménage (?)

1.4 ilbm_new

ilbm.m/ilbm_New ilbm.m/ilbm_New

Syntaxe

```
ilbmhandle := ilbm_New( nom:PTR TO CHAR,
                        drapeau:LONG );
```

Description

Crée une structure de gestion (privé) ilbm, et repli plusieurs champs. Le fichier spécifié par 'nom' est ouvert, et les chunks IFF ILBM BMHD, CAMG, et CMAP sont analysé (parsed).

EntréeS

nom Une chaine terminée par NULL, spécifiant le nom de l'image. Elle DOIT être présente!

drapeau Un masque d'options, qui sont :

ILBMF_COLOURS4
crée une version compatible à LoadRGB4() de la palette, et met un pointeur dessus dans le bloc info de l'image comme 'pal4' (voir ilbm_PictureInfo()

ILBMF_COLOURS32
crée une version compatible à LoadRGB32() de la palette, et met un pointeur dessus dans le bloc info de l'image comme 'pal32' (voir ilbm_PictureInfo()

SortieS

ilbmhandle Un gestionnaire !!PRIVÉ!! utilisé avec les autres fonctions ilbm_xxx. Si pour quelques raisons ça ne marche pas, ce sera 0.

Notes

Voir aussi

ilbm_Dispose(), ilbm_LoadPicture(), ilbm_PictureInfo()

1.5 ilbm_dispose

ilbm.m/ilbm_Dispose ilbm.m/ilbm_Dispose

Syntaxe

```
ilbm_Dispose( iffhandle:LONG );
```

Description

Ferme le fichier original, libère les affaires de l'iffparse.library, et ferme quelques bibliothèques. Utilisez ça pour libérez des données

```
       inutile une fois que l'image a été chargé.

 EntréeS
       iffhandle       le gestionnaire obtenu par ilbm_New(), ou 0.

 SortieS

 Notes
       Il vaut mieux passer iffhandle:=0 à cette fonction.

 Voir aussi
       ilbm_New()
```

1.6 ilbm_pictureinfo

```
ilbm.m/ilbm_PictureInfo                                    ilbm.m/ilbm_PictureInfo

 Syntaxe

       pictureinfo := ilbm_PictureInfo( iffhandle:LONG )

 Description
       Retourne un pointeur sur un objet pictureinfo qui contient des infos
       sur l'image chargée.

 EntréeS
       iffhandle       Un gestionnaire VALIDE obtenu par ilbm_New().

 SortieS
       pictureinfo     Un pointeur sur un objet de type 'pictureinfo'.
               Les champs vont être mis comme suit:

               bmhd    pointeur sur le BitMapHeader du fichier IFF
               modeid  le modeid, obtenu du chunk CAMG - ou 0.
                       Il peut aussi être fixé par l'application avant
                       d'appeler ilbm_LoadPicture()
               colours nombre de couleurs représentés dans l'image.  Une
                       image IFF-24 aura 16,777,216 placé ici!
               palraw  Si le nombre de couleurs (dessus) est de 256 ou moins,
                       et un chunk CMAP éxiste, palraw est un pointeur sur
                       une palette 24-bit raw lu du fichier IFF. Les couleurs
                       sont placées par groupes de 3 octets - rouge/Vert/Bleu
               pal4    Si ILBMF_COLOURS4 était spécifié quand le gestionnaire
                       (iffhandle) a été créé, et il y avait un CMAP, ET il y
                       avait assez de mémoire, pal4 est un pointeur sur un
                       tableau compatible LoadRGB() de couleurs
               pal32   Si ILBMF_COLOURS32 était spécifié quand le gestionnaire
                       (iffhandle) a été créé, et il y avait un CMAP, ET il y
                       y avait assez de mémoire, pal32 est un ointeur sur un
                       tableau compatible LoadRGB() de couleurs.

 Notes
       Le champ modeid est le seul qui peut être écrit (writeable)! Tous les
       autres ne peuvent être que lu (read-only)
```

```
       Si la mémoire est courte, les champs pal4 et pal32 peuvent rester à
       0, même si il était demandé avant. Ce srait bien de toujours de
       vérifier ces champs avant utilisation.

 Voir aussi
       ilbm_New(), ilbm_Dispose(), ilbm_LoadPicture()
```

1.7 ilbm_loadpicture

```
ilbm.m/ilbm_LoadPicture                                    ilbm.m/ilbm_LoadPicture

 Syntaxe

       status := ilbm_LoadPicture ( iffhandle:LONG,
                                    taglist:LONG )

 Description
       Charge l'image dans l'environnement spécifié.

 EntréeS
       iffhandle       Un gestionnaire (iffhandle) obtenu utilisant ilbm_New(),  ↩
          ou 0
            dans chaque cas ou une erreur sera rencontré.
       tags    Une tag-list spécifiant les options de chargements.  Actuellement
            les tags définis sont:

               ILBML_BITMAP    tag.data pointe sur un bitmap existant dans
                               lequel on mettra les données de l'image. Le
                               bitmap a besoin d'être assez grand ...
               ILBML_SCREEN    tag.data pointe sur un écran ouvert dans
                               lequel la palette/image sera chargée.
               ILBML_CHUNKY    tag.data spécifie un tableau d'octets pour
                               mettre la version chunky-pixel de l'image
                               PAS IMPLEMENTE
               ILBML_GETBITMAP Ceci spécifie que ilbm_LoadPicture() allouera
                               son propre bitmap.  Dans ce cas tag.data est
                               un pointeur sur une variable qui aura le bitmap
                               voulue.
                               ilbm_FreeBitMap() DOIT être utilisé
                               pour libérer le bitmap.
               ILBML_GETSCREEN spécifie que ilbm_LoadPicture() ouvrira
                               l'écran pour vous. tag.data pointe sur une
                               variable qui aura lz pointeur d'écran une fois
                               obtenu. Si l'écran ne peut être ouvert, 0 est
                               mis dans cette variable. L'écran doit être
                               fermé par CloseScreen() - Ca peut se passer
                               après avoir appelé ilbm_Dispose().
               ILBML_GETCHUNKY Deninez!  PAS IMPLEMENTED
               ILBML_SCREENTAGS si ILBML_GETSCREEN est utilisé;, alors ce tag
                               peut être utilisé pour spécifier de nouveaux
                               tags à l'ouverture d'écran. Les tags suivants
                               NE peuvent PAS être utilisé : SA_WIDTH,
                               SA_HEIGHT, SA_DEPTH, SA_DISPLAYID.
               ILBML_NOCOLOUR  si SA_SCREEN/SA_GETSCREEN a été spécifié,
                               alors utiliser ce tag booléen préviendra
```

```
                                ilbm_LoadPicture() de mettre la palette pour
                                l'écran.  A n'utiiser que seulement si c'est
                                vrai.

 SortieS
        status  =0 si tous a bien marché, ou négatif pour les erreurs.
                   (voir ilbmdefs.m)

 Notes
        Rappelez vous, si un des tags 'GET' a été utilisé, c'est à l'application
        de libérer ce qu'il y a eu.

 Voir aussi
        ilbm_New(), ilbm_Dispose(), ilbm_PictureInfo(), ilbm_FreeBitMap()
```

1.8 ilbm_freebitmap

```
ilbm.m/ilbm_FreeBitMap                                  ilbm.m/ilbm_FreeBitMap

 Syntaxe

        ilbm_FreeBitMap( bitmap )

 Description
        Libère un bitmap retourné par ilbm_LoadPicture(), via le tag
        ILBML_GETBITMAP.

 EntréeS
        bitmap  un bitmap VALIDE comme celui retourné par le tag
                ILBML_GETBITMAP.

 SortieS

 Notes
        Si V39 est présente, cette fonction appelle juste FreeBitMap() -
        sinon elle utilise ses propres routines.

 Voir aussi
        ilbm_LoadPicture()
```

1.9 Information sur les exemples

```
 Cette section décrit les exemples sources.

 showpic

        Un simple afficheur d'image. Il montre un façon facile de charger et
        afficher une image sur un écran Amiga. L'utilisation de l'asl.library
        y est aussi à l'honneur.

        usage:
                showpic
```

picwindow

Un autre simple afficheur d'image. Celui-ci affiche l'image sur l'écran workbench, dans une fenêtre de bonne taille. Il montre le chargement dans les bitmaps, l'obtention des informations sur l'image avant l'affichage, et 'blitting' dans les fenêtre workbench.

usage:

```
picwindow
```