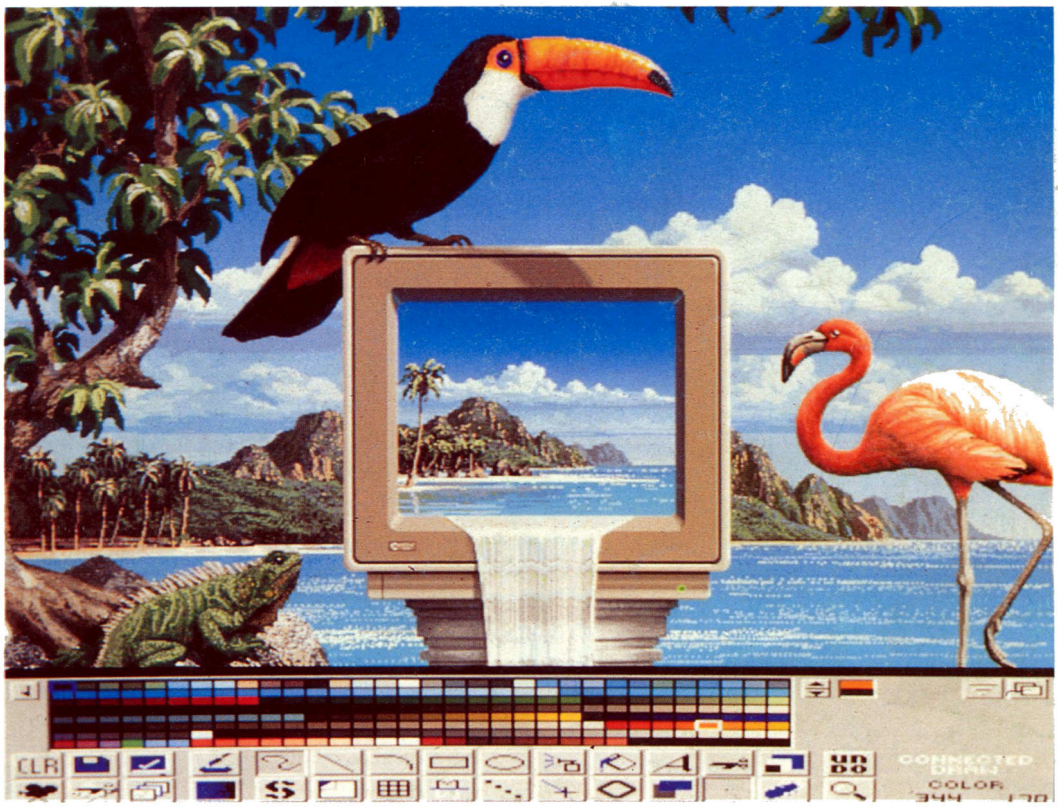
ISSN 1164-1746

AmigaNews

L'ORDINATEUR CREATIF



PC OU Amiga? - Le début de notre grand débat-

Expo: World of Amiga Frankfurt

Test: Amiga 1200, Carte OpalVision, Lecteur CD A570, Sauvegarde sur Vidéocassette, A3000 UNIX, Accords 2000, UIK, Pixel 3D, Interchange Plus, Scenery Animator.... et beaucoup plus

M 4584 - 53 - 30,00 F

JANVIER 1993 No. 53 Suisse 9.40 FS, Belgique 219FB, Canada $5.75

UIK et la Programmation Orientée Objet

Lisez attentivement ce qui va suivre car la notion de POO (pour Programmation Orientée Objet...) est non seulement à la pointe du développement informatique mais de plus, elle entre en force sur notre machine préférée. Citons trois produits qui utilisent actuellement cette notion de POO : UIK (User Interface Kit), le c++ et... le system 3.0! Le system 3.0? La "Datatypes.library" exploite en effet les atouts de la POO. Merci monsieur Commodore..

1. Origines de la POO

Au départ il y a bien entendu le langage C. Celui-ci reste un outil de développement très puissant permettant de manipuler à la fois des concepts de haut et de bas niveaux (cf assembleur). Mais il ne suffit pas de pouvoir bénéficier d'un langage de haut niveau pour résoudre d'un coup de baguette magique les problèmes rencontrés par le génie informatique. Restent en effet les éternels problèmes de maintenance et de réutilisation des applications.

Pour bien comprendre, vous devez vous rendre compte que plus un programme est modulaire et indépendant des données qu'il utilise, plus sa maintenance par exemple en sera facilitée. La modularité ne peut pas non plus tout résoudre car chaque module d'un programme C ne peut communiquer avec un autre qu'au travers d'une interface externe réduite, l'intérieur de ces modules restant une sorte de boîte noire inviolable par les autres modules... On parle alors de phénomène d'ENCAPSULATION (limitation d'accès à l'information).

Lorsque vous écrivez deux programmes, utilisant une interface graphique de haut niveau, il vous faudra certainement revoir vos routines d'affichage de menus (par exemple) qui devront être parfaitement adaptées à chaque application. Le morceau de choix de la programmation objet se base sur une nouvelle nécessité qui est la REUTILISATION des modules/données. En effet, l'analyse de l'évolution des logiciels a montré, qu'en général, les éléments les plus stables étaient les structures de données et que les modifications portaient essentiellement sur les fonctions de manipulations de ces informations.

Cette analyse a conduit à l'idée de CENTRALISER les données en leurs ASSOCIANT les traitements spécifiques qui les manipulent plutôt que de centrer les programmes sur les fonctions, traitements qui sont susceptibles d'être modifiées plus fréquemment. Il s'agit du concept de base, fondamental, de la POO: ASSOCIER LES TRAITEMENTS AUX DONNEES.

2. Objectifs de la POO

a) Rendre l'application portable

C'est la possibilité d'utiliser le programme source sur d'autres machines ou environnements différents. La POO permet d'isoler les détails d'implémentation aux niveaux les plus bas, c'est à dire lors de la spécificité des objets eux-mêmes. Donc pour rendre une application portable en POO il suffit de redéfinir les données décrites des objets utilisés. (cfr Datatypes.library kickstart 39)

b) La compatibilité

Mon application doit s'intégrer dans l'environnement existant. C'est à dire qu'il ne suffit pas qu'elle fonctionne correctement. Elle ne doit pas empêcher le déroulement normal d'autres tâches en cours. Bref, elle doit respecter le multi-tâche. (cfr UIK)

c) La vérifiabilité

C'est sous ce terme que l'on désigne la manière plus ou moins aisée de définir des procédures testant les logiciels. La modularité de la POO permet une vérification plus aisée, élément par élément.

d) La réutilisabilité

Pour pouvoir réutiliser des modules développés pour une autre application il faut fournir un effort dès la conception afin de produire des modules "paramétrables". Ces modules doivent faire face à toutes les situations que l'on pourrait rencontrer ultérieurement. L'emploi de L'HERITAGE (cf plus loin) en POO apparaît comme une solution très encourageante.

e) L'extensibilité

C'est la faculté de pouvoir ajouter des données ou/et modules supplémentaires sans endommager le logiciel existant. Les notions de limitation d'accès et d'héritage y répondent partiellement.

On ne peut pas tout avoir... Par exemple, un programme efficace (tenant compte de toutes les possibilités de la machine) sera très difficilement portable... Car un programme portable se cantonne à formuler un concept de manière standard et est donc bien souvent peu performant.

Dans l'ensemble la POO donne plus de réponses au "génie informatique" [un copain à moi :-)] que la programmation dite classique.

3. Notion de CLASSE

Définition: "Entité qui définit de manière générale la structure, les propriétés et les opérations d'éléments de données manipulés par le logiciel". A la base de la notion de classe se trouve le concept de TYPE ABSTRAIT... Pour ne pas rentrer dans trop de détails techniques voici un exemple de type abstrait:

```
struct rectangle
{
   float longueur, largeur;
   float perimetre()
   {
      float perim;
      perim = 2 * (longueur + largeur );
      return(perim);
   }
};
struct rectangle rect;
```

La structure rectangle contient une entrée de type "float" qui sera donnée comme étant la valeur retour de type "float" de la fonction "perimètre"! ON ASSOCIE AUX DONNEES LES FONCTIONS QUI LES MANIPULENT!

En POO les éléments constituant la structure sont appelés des MEMBRES. Une fonction membre est appelée METHODE.

La déclaration de la fonction se fait à l'intérieur même de la structure définissant le type abstrait. Il est tout à fait concevable de se contenter de définir le PROTOTYPE de la fonction seulement (La fonction est alors définie plus loin dans le source).

La structure "rectangle" est une classe. "rect" est un objet.

Pour appeler la fonction "perimètre" il faut utiliser l'instruction "rect.permimetre()" pour envoyer le message "perimetre" à l'objet "rect" de classe "rectangle", ce qui provoque l'appel de la méthode "perimetre()".

4. L'héritage

Le principe de l'héritage est de décrire un nouveau type abstrait non pas à partir de rien mais par extraction d'un type (ou plusieurs) déjà existants. Hériter d'une classe consiste à recevoir toutes les caractéristiques de cette classe et à y adjoindre des éléments supplémentaires. En gros, lorsqu'une classe X hérite d'une classe Y, la classe héritière est riche de toutes les fonctions offertes par la classe parente (ou mère) Y. En ce qui concerne les données, la classe fille possède toutes les données de la classe mère. En d'autres termes, la classe héritière possède tous les membres (données et fonctions) de la classe mère. Les nouveaux membres hérités peuvent aussi être redéfinis et on obtient alors la notion d'héritage en cascade...

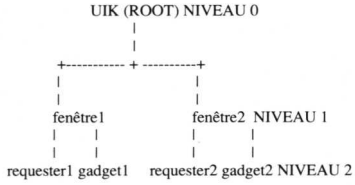
5. UIK

UIK est l'abréviation de User Interface Kit. Il s'agit d'une "library" système qui reprend le concept général de la POO. UIK est livré avec un manuel de 88 pages d'aspect assez technique. Je pense que même si évidemment UIK peut être utilisé avec n'importe quel langage compilé il est préférable d'avoir une solide connaissance en C avant de se plonger tête baissée dans le manuel... Dès le départ l'auteur nous parle d'objet et si vous n'avez aucune notion de POO cela risque de vous paraître assez étrange à première vue.

UIK fonctionne sur n'importe quel AMIGA et sous n'importe quel kickstart. L'uik.library vous privera d'un volume de 60K dans votre LIBS:

Objet UIK

Le noeud (root) de base est l'objet UIK. A partir de celui-ci on construit en cascade d'autres objets. La notion d'héritage est ici une réalité: si on supprime un parent tous ses fils seront également supprimés.



UIK, fenêtre1, fenêtre2, requester1, gadget1, gadget2 sont des OBJETS. L'objet requester1 disparaît si on enlève l'objet fenêtre1. L'objet gadget1 disparaîtra à son tour car il est fils de fenêtre1.

Les objets sont de deux types. Les objets INTERNES et les objets EXTERNES. Un objet interne est disponible dès l'ouverture de "l'UIK. library". Un objet externe se trouve dans un répertoire (UIK:Objects), UIK devant le charger en mémoire avant exécution. Une documentation complète est disponible décrivant les objets internes et externes.

Les étapes de base d'un programme utilisant UIK.library sont:

1) Ouvrir l'UIK.library
2) Ajouter les objets
3) Ajouter un événement à attendre
4) Activer les objets
5) Attendre un événement utilisateur
6) désactiver les objets
7) fermer l'UIK.library.

Le gain de temps d'une telle programmation est énorme! Le programmeur est enfin déchargé d'une programmation lourde qui peut se révéler peu fiable.

1) Ouvrir l'UIK.library

Rien de très sorcier:

```
struct UIKBase *UIKBase = NULL;
void main(void) {
if(UIKBase = struct UIKBase *) OpenLibrary(UIK.library",0L)) {
...
}
else printf("ERREUR: J'ai un besoin vital de la LIBS:UIK.LIBRARY !");
}
```

2) Ajout d'objets

L'ajout d'un objet ne signifie pas son activation. Cela revient à une déclaration de l'objet. On ajoute un objet en le reliant à son parent. L'objet primaire (ROOT) UIK ne possède pas de parents. C'est un objet spécial qui porte le nom de "graine": tous les autres objets sont construits à partir de celui-ci.

```
UIK = (struct UIKGlobal *) UIK_AddObject("UIK",0,TagList);
gadget1 = UIK_AddObject("UIKObj_Boolean",fenêtre,TagList);
```

3) Ajout d'événements

A chaque objet on peut associer un événement (ou plusieurs). Ce couple objet/événement est appelé VECTEUR. Si l'événement se produit UIK exécute la fonction qui lui correspond en lui transmettant des paramètres. UIK prend en charge la gestion d'attente d'événements, étant toujours prêt à recevoir un événement utilisateur.

```
UIK_AddVector( gadget1,action1,GADGETUP,requester);
```

4) Activer les objets

Si un objet parent est activé, il activera en cascade ses fils. Donc l'écriture:

```
UIK_Start(UIK); réalise l'activation de TOUS les objets.
```

Un nom d'objet seul peut-être spécifié mais si cet objet possède des enfants ils seront activés à leur tour.

5) Attendre un événement utilisateur

```
UIK_Do(UIK,SIGBREAK_CTRL_C); boucle UIK tant que
                             ; les touches CTRL et C
                             ; ne sont pas d'actualité.
```

Attention on peut également envisager le schéma suivant: nous ouvrons une fenêtre avec un gadget. L'activation du gadget nous branche sur action1() qui est une fonction utilisateur. Action1() recherche un pointeur sur sa propre tâche et envoie le signal CTRL_C, résultat: le programme se saborde lui-même...

```
void action1(void) { Signal(FinTask(0),SIGBREAK_CTRL_C); }
```

6-7) Désactiver les objets et fermer la librairie

Ceci est réalisé très simplement, il suffit de désactiver l'objet de base, c'est à dire la graine UIK. Tous les objets y étant réliés (ils sont tous des enfants) ils seront ainsi désactivés.

```
if(UIK)      UIK_Remove(UIK);
if(UIKBase)  CloseLibrary(UIKBase);
```

6. Exemple concret

Je n'ai eu de problème particulier à utiliser DICE. Bien entendu aucun problème avec le SAS car c'est le compilateur qu'utilise Jean-Michel.

UIKShow utilise trois objets externes: l'OBJILBM, OBJSelectFile et l'OBJjoystick. Dommage que l'objet MOUSEJOY soit en cours de développement car il aurait pu servir dans cet exemple. La LIBS:UIK.library est LIBREMENT distribuable (ainsi que les objets). Vous pouvez donc vous la procurez dans le domaine public. (Mais si vous ne la trouvez pas, allez voir dans le Petit Amiga Illustré du mois de Janvier 93, elle s'y trouve).

```
/*------------------------------------------------------------------- *
*          /\/\/\/\                                                  *
*          / UIKShow.c /                                             *
*          \/\/\/\/                                                  *
*                                                                    *
* = PROGRAMME D'EXEMPLE DE VISUALISATION =                           *
* = D'UNE IMAGE IFF GRACE A L'OBJET EXTERNE =                        *
*                 =  "OBJILBM" D'UIK. =                              *
*                                                                    *
*     Sous DICE : dcc UIKShow.c UIK.lib -o UIKShow                   *
*     ~~~~~~~~~                                                      *
*------------------------------------------------------------------- */

#include <stdio.h>
#include <libraries/dos.h>
#include <intuition/intuition.h>
#include "uiki:uikbase.h"
#include "uiki:uikobj.h"
#include "uiki:uikmacros.h"
#include "uiki:uikglobal.h"
#include "uiki:uik_protos.h"
#include "uiki:uik_pragmas.h"
#include "uiki:objfileselector.h"
#include "uiki:objscreen.h"
#include "uiki:objwindow.h"
#include "uiki:objilbm.h"
#include "uiki:objjoystick.h"
/*------------------------------------------------------------------*/
#define WR(str) fwrite(str,strlen(str),1,stdout)
#define VERSTAG  "\0$VER: UIKShow 1.00 (15-11-92)"
/*------------------------------------------------------------------*/

struct    UIKBase          *UIKBase   = NULL;
struct    UIKGlobal        *UIK       = NULL;
struct    UIKObj           *ObjILBM   = NULL;
struct    UIKObj           *Ecran     = NULL;
struct    UIKObj           *Entree    = NULL;
struct    UIKObjWindow     *wo        = NULL;
struct    UIKIlbmFrame     UIKFrame;
FILE      *fp = NULL;
ULONG     *PtrVM,VM;
char      Buffer[512], *PtrBuff=Buffer, *Cherche, NF[128], *PtrNF = NF;
BOOL      FlagSelect;
void      CopyRight(void);
void      LibereTout(void);
void      Selection(struct UIKObjFileSelector *fso,UBYTE *filename);
void      CancelFonc(void);
void      action(void);
void      main(void);
/*------------------------------------------------------------------*/
struct    TTextAttr WinTTA = {"Times.font"
                              ,24,FSF_ITALIC|FSF_UNDERLINED,0,0};
struct    TagItem  wtl[] = {
          {  UIKTAG_OBJ_LeftTop,SETL(158,18) },
          {  UIKTAG_OBJ_WidthHeight,SETL(318,170) },
          {  UIKTAG_OBJ_Title,1 },
          {  UIKTAG_OBJWindowFl_With_Close,FALSE },
          {  UIKTAG_OBJ_AltTitle,1 },
          {  UIKTAG_OBJ_ActInactPens
                              ,UIKCOLS(UIKCOL_LIGHT,UIKCOL_GREY,
             UIKCOL_WHITE,UIKCOL_BLACK) },
          {  UIKTAG_OBJWindow_MinimumWidthHeight,SETL(200,150) },
          {  UIKTAG_OBJ_TTextAttr,(ULONG)&WinTTA},
          {  TAG_END } };
struct    TagItem  fstl[] = {
          {  UIKTAG_OBJ_LeftTop,SETL(7,3) },
          {  UIKTAG_OBJ_WidthHeight,SETL(300,165) },
          {  UIKTAG_FS_OKFUNC,(ULONG) Selection },
          {  UIKTAG_FS_CANCELFUNC,(ULONG) CancelFonc },
          {  UIKTAG_OBJ_FontName,(ULONG) "topaz.font" },
          {  UIKTAG_OBJ_FontHeight,(ULONG) 8 },
          {  TAG_END } };
/*------------------------------------------------------------------*/
UBYTE   *LangEnglish[] =
{
   "",
   "UIKShow",
   0
};
UBYTE   *LangOther[] =
{
   "",
   "UIKShow",
   0
};
/*------------------------------------------------------------------*/
void     CopyRight(void)
{
   WR("\n"\
      "\t /\/\/\/\/\/\/\\n"\
      "\t/  UIKShow   / V1.0 AUTEUR : LECLERCQ XAVIER\n"\
      "\t\/\/\/\/\/\/\/          Vieux chemin d'ath n°12\n"\
      "\t                  B-7548 Warchin BELGIQUE\n\n"\
      "» A l'affichage il faut appuyer sur FIRE ou CTRL-C pour sortir...\n\n");
}
/*------------------------------------------------------------------*/
void     LibereTout(void)
{
   if (Ecran)       UIK_Stop(Ecran);
   if (ObjILBM)  UIK_CallObjectFunc(ObjILBM, UIKFUNC_ILBM_CleanupFrame\
                        ,(ULONG)Ecran,(ULONG)&UIKFrame);
   if (UIK)         UIK_Remove(UIK);
   if (UIKBase)     CloseLibrary(UIKBase);
   exit(0);
}
/*------------------------------------------------------------------*/
void     Selection(struct UIKObjFileSelector *fso,UBYTE *filename)
{
   FlagSelect = FALSE;
   strcpy(PtrNF,filename);
   Signal(FindTask(0),SIGBREAKF_CTRL_C);
}
/*------------------------------------------------------------------*/
void     CancelFonc(void)
{
   Signal(FindTask(0),SIGBREAKF_CTRL_C);
   FlagSelect = TRUE;
}
/*------------------------------------------------------------------*/
void     action(void)
{
   Signal(FindTask(0),SIGBREAKF_CTRL_C);
     }
/*------------------------------------------------------------------*/
void     main(void)
{
   CopyRight();
   if (UIKBase = (struct UIKBase *) OpenLibrary("uik.library",0L))
   {
        if (UIK = (struct UIKGlobal *) UIK_AddObjectTags( "UIK",0
             ,UIKTAG_GEN_LangEnglish,(ULONG) LangEnglish
             ,UIKTAG_GEN_LangOther,(ULONG) LangOther,TAG_END))
        {
          if (wo = (struct UIKObjWindow *)
             UIK_AddObject("UIKObj_Window",UIK,wtl))
          {
            if (UIK_AddObject("UIKObj_FileSelector",wo,fstl))
            {
              UIK_Start(UIK);
              UIK_Do2(UIK,SIGBREAKF_CTRL_C,1);
            }
            else
              WR("ERR 09: Problème pour charger l'UIK:UIKObj_FileSelector !??\n"\
                 "~~~~~~~\n\n"),LibereTout();
          }
          else
            WR("ERR 08: Problème pour ouvrir une fenêtre !??\n"\
               "~~~~~~~\n\n"),LibereTout();
        }
        else
          WR("ERR 02: Le segment 'root' UIK ne veut pas s'initialiser !??\n"\
             "~~~~~~~\n\n"),LibereTout();
   }
   else
        WR("ERR 01: J'ai un besoin vital de la LIBS:UIK.LIBRARY !??\n"\
           "~~~~~~~\n\n"),LibereTout();
   if (UIK)         UIK_Remove(UIK),UIK = NULL;
   if (UIKBase)     CloseLibrary(UIKBase),UIKBase = NULL;
   if (FlagSelect)  exit(0);
   VM = 0; /* Si le fichier IFF ne possède pas un CAMG ?!.. */
   if(!(ferror(fp=fopen(NF,"rb"))))
   {
        fread(PtrBuff,1,sizeof(Buffer),fp);
        for (int i = 0; i < sizeof(Buffer) ; i++ , PtrBuff++)
        {
          if (   *(PtrBuff + 0)  == 'C' &
                 *(PtrBuff + 1)  == 'A' &
                 *(PtrBuff + 2)  == 'M' &
                 *(PtrBuff + 3)  == 'G')
          {
            PtrVM = (PtrBuff + 8);
            VM    = *PtrVM;
          }
        }
        fclose(fp);
   }
   else
        WR("ERR 07: Je n'arrive pas à ouvrir le fichier IFF !??\n"
           "~~~~~~~\n\n"),exit(0);
   if (UIKBase = (struct UIKBase *) OpenLibrary("uik.library",0L))
   {
        if (UIK = (struct UIKGlobal *) UIK_AddObject( "UIK",0,0))
        {
          if (ObjILBM = UIK_AddObject( "UIKObj_ILBM", UIK , 0))
          {
            if (IFF_DONE == UIK_CallObjectFunc(ObjILBM
                             ,UIKFUNC_ILBM_File2BitMap,\
              (ULONG)PtrNF,(ULONG)&UIKFrame))
            {
              if (Ecran = (struct UIKObj *) UIK_AddObjectTags("UIKObj_Screen"\
```

```
                    ,UIK
                    ,UIKTAG_OBJ_WidthHeight
                    ,SETL(UIKFrame.W,UIKFrame.H)
                    ,UIKTAG_OBJScreen_ShowTitle,FALSE
                    ,UIKTAG_OBJScreen_ScreenQuiet,TRUE
                    ,UIKTAG_ObjScreen_ViewModes,VM
                    ,UIKTAG_ObjScreen_CustomBitMap
                    ,(ULONG)UIKFrame.BMap
                    ,TAG_END))
            {
              if ( Entree = (struct UIKObj *)UIK_AddObjectTags(
                    "UIKObj_joystick"
                    ,Ecran
                    ,UIKTAG_Joystick_LButFunc,action
                    ,UIKTAG_Joystick_RButFunc,action
                    ,UIKTAG_Joystick_MButFunc,action
                    ,TAG_END))
              {
                if (UIK_Start(UIK))
                {
                  UWORD *MouseOFF = (UWORD *)0x00DFF096;
                  *MouseOFF = 0x20;
                  UIK_CallObjectFunc(ObjILBM
                        ,UIKFUNC_ILBM_Colors2Screen\
                        ,(ULONG)Ecran,(ULONG)&UIKFrame);
                  UIK_Do2(UIK,SIGBREAKF_CTRL_C,1);
                }
                else
                  WR("ERR 06: Problème avec UIK_Start !??\n"\
                    "~~~~~~~\n\n");
              }
                else
                  WR("ERR 10: Problème avec
                    l'UIK:UIKObj_Joystick !??\n"\
                    "~~~~~~~\n\n");
            }
            else
              WR("ERR 05: Problème pour ouvrir l'écran !??\n"\
                "~~~~~~~\n\n");
          }
          else
            WR("ERR 04: Problème de la routine
              de chargement IFF !??\n"\
              "~~~~~~~\n\n");
        }
        else
          WR("ERR 03: Je ne trouve pas l'UIK:UIKObj_ILBM !??\n"\
            "~~~~~~~\n\n");
      }
      else
        WR("ERR 02: Le segment 'root' UIK ne veut pas s'initialiser !??\n"\
          "~~~~~~~\n\n");
  }
  else
    WR("ERR 01: J'ai un besoin vital de la LIBS:UIK.LIBRARY !??\n"\
      "~~~~~~~\n\n");
  LibereTout();
}
/*--------------------------------------------------------------------------*/
```

7. Conclusion

En ne prenant pas en compte la période d'adaptation au logiciel, j'ai programmé UIKShow très rapidement, ceci grâce à la programmation objet et surtout aux objets existants déjà! Il manque à UIK un peu de finition globale et quelques objets supplémentaires mais dans l'ensemble je pense qu'UIK répond à ce que l'on attend de lui. La POO ne résoud pas tous les problèmes du génie informatique mais réduit le temps de développement. Que demander de plus? C'est un véritable plaisir de programmer (correctement) avec UIK. Pour moi il s'agit de l'utilitaire d'aide à la programmation de l'année...

Xavier Leclercq

Vieux Chelin d'ath n°12

B-7548 Warchin (BELGIQUE)

Logiciel:	UIK
Version testée:	1.1
Manuel et docs:	en francais
Auteur:	JM Forgeas
Prix:	550 FF
Distributeur:	JM Forgeas 16170 BordeVille, France

Accessibilité	: 7/10
Prise en main	: 7/10
Efficacité	: 9/10
Rapidité	: 9/10
Convivialité	: 9/10
Rapport Qualité-prix	: 9/10
Note globale	: 9/10

PETITES ANNONCES

de bastian 834500 La Seyne

Vds A2000 ROM 20 avec 3Mo deRAM DD Quantum 52 Mo (17 ms), lect. supp. 3", DD 32 Mo + lect. 5" + carte série sur carte PC-XT, compilateur C Lattice V5.04 (original + docs), bible de l'Amiga + autres livres de programation C sur l'Amiga, Amiga C Encyclopédia (disks + sorties sur papier), jeux "croisière pour un cadavre" (original), nombreux disks: le tout cédé à 7500F. Tel au 81.98.18.15 ou Fax 85.59.05.43 (destinataire: ALF) Je suis souvent dans les régions Montbéliard et de Cluny

Recherche collaborateurs bénévoles pour réalisation d'articles sur Amiga. En vue de finir un magazine sur disquette. Vous pouvez envoyer vos oeuvres (Articles en français au format ASCII, dessins...) à Bail sylvain (TSD: CARTEL) + 5 ave Louis Bleriot 44340 Bouguenais. ps: le premier numéro sera envoyé à chaque participants.

Vds DD pour A1200 ou A600 120 Mo: 1800F; 80Mo: 1400F. Pour A4000: 210Mo, IDE: 2600F. Laurent Tel: (1) 48.68.21.84

Vds A2000B, 1.3 et 2.0, Moniteur 1084S, DD 50 Mo 6000F, Genlock A2300 800F 2ième lecteur 400F. Action Replay MK III 400F. Tel: 20.93.31.07 Romuald.

Poour A 1.3.2.0 Vds Scanner PowerScan 400 dpi avec log. 1090F: Maxi Plan IV 400F, DémoMaker 250F, PPage 2.0 1400F, Excellence 2.0 500F, Sonix 20 300F. Volumn 4D 350F, Scala 500 500F. Tous ces logiciels originaux avec docs en français et boite d'origine. Cause Amiga 1200 Sadoine Jean 76 Rue d'Hurlupin 59560 Comines. Tel: 20.39.11.52

Vds carte graphique 16 millions de couleurs VD - 2001 contenant un digitaliseur en temps réel et un genlock + logiciel VD Paint (acheté Nov. 92!). Prix sacrifié: 13600F! Vds égalamant LATTICE C v.51 (800F). Tel: 66.84.44.92 ap 20h.

Vds AVidéo 12 + logiciels: 1500F, Scanner GoldenImage: 600F, ProDraw 2: 500F, PowerWoork: 3logiciels bureau: 400F, TurboSilver: 400F, Sculpt 3Dxl: 600F, DigiView Good 4 + DigiPaint3: 600F, Real 3D 1.4 Pro: 2000F, tous originaux avec docs. Tel: 82.85.75.48

Vds A500 (1990) + lecteur externe + extension mémoire A580 (1.5 Mo et horloge) + Atoonce - Amiga (émulateur PC) + DD (80 Mo) + nmbreux logiciels originaux (langage, vidéo, jeux) + nombreux livres 5000F à débattre. Tel: (Lyon) 78.28.71.39 le soir.

Vds plan, circuit imprimé, logiciels pour réalisation d'un Digitalisateur audio pour Amiga, 85F P.Godin 5 rue Berthelot 60570 Andeville.

Urgent! Vds un Amiga 500, ROM 1.3, 2 Mo de RAM totale + horloge et switch + ses 3 dg d'utilisation + la souris et son tapis + une imprimante Citizen 120-D avec docs. Possibilité de booter en DF1: optionnel. Le tout pour 3200F ou séparement à discuter. Je vends également extension 512Ko Switch et horloge pour A500: 250F. Intéressé? Appelez Nicolas au 60.08.89.12

Vds A2000B WB 2.0 et 1.3 + 1Mo RAM vidéo + 6 Mo RAM FAST + DD 40 Mo + genock A 2300 + imprimante MPS 1000 + originaux: DP3, Photon2, Amos... 10000F ou 11000F avec écran 103 stéréo. Vente sur Brest - Vannes - Paris Jean Loouis au 98.47.64.68 e week end ou laisser un méssage en semaine.

Vds A2000 2.0/ 1.3 (sous garantie) + 1084S + 9Mo + 2iéme lecteur interne + 2 lecteurs externes + A.Replay MK III + Sampler MKII + imprimante Star LC 10 couleur + DD GVP HCD 52 Mo + cyclone copieur hard + quadrupleur de joysticks + nombreux logiciels + émulateurs PC, MAC, ST + 2 joysticks + papier imprimante + 250 disquettes de softs! (crossdos, dos2dos.) le tout 15000F Tel: 49.11.19.15 ou laissez votre TEL (répondeur).

Vds 4Mo RAM static column A3000: 1800F. Vds AVidéo 24 + TVPaint: 5000F ou échange contre syquest 8 Mo + 1 cartouche.

Les anciens numéros sont disponibles chez:

France

13 **Infologs** 205 Rue ST Pierre 13005 Marseille
31 **Infonix**, 12 &14 rue Réclusane 31300 Toulouse
31 **Volumm** 30 r Pharaon 31000 Toulouse
62 **Microtech** 32B r Florent Evrard 62420 BillyMontigny
62 **Softone**, 394 rue de Lille, 62400 Bethune
64 **Bab Micro**, 7 rue de Coursic 64100 Bayonne
69 **Gelain Ets** 22 Ave de Saxe, Lyon
75 **Phase**, 93 av du Gal Leclerc 75014 Paris
91 **Essonne Mailing** 8 rue du Bois Sauvage 91024 Evry

Québec

Maison du Logiciel,2466 J-Talon Est,Montreal H2E1W2
Info Plus 1828 Rue Notre Dame, Trois Rivieres G9A4Y1
Visipro 991, Boul. Talbot Chicoutimi G7G 3W5

Belgique

Media Lem, r Francois Dorzee 93, 7360 Boussu
MiA Software (voir ci-dessus)

Suisse

10 **Distrib. Electronique** 24 av de Cour 1007 Lausanne
10 **Mix Image,** Av. de France 60 1004 Lausanne
12 **Edu Soft** 14-16 r des Gares 12011 Genève 2
12 **Distrib. Electronique** r Vollandes 62, 1207 Genève
14 **M.J.S. Informatique**, Pl Pestalozzi 9, 1400 Yverdon
19 **-I- Computer**, Grd Pont 33 CH 1950 Sion
20 **Octopus**, r du Bassin 8, 2000 Neuchatel

Le Petit Amiga Illustré

Sommaire du Numéro # 4 - Janvier 93

007 : Exemple de programmation d'un cheval de Troie dans le cadre d'une étude au sujet des fléaux informatiques. Ce programme ne détruit aucune information , n'appporte aucune nuisance à l'utilisateur , et ne se reproduit pas. Ce 'mouchard' informatique a été programmé dans un cadre purement didactique pour analyser le comportement des 'piégés' et des 'piégeurs'. Source assembleur compris.
Amos : Programmation d'un "ScrollText" en AMOS. Les explications de ce tour de force démontrant la puissance de ce langage vous sont données de long en large. Source AMOS compris.
SSP : Système_Solaire_Poids est un utilitaire qui calculera votre poids vis à vis des différentes planètes du système solaire. Simple mais instructif... Source C compris.
XMPrint : Permet l'impression d'un fichier ASCII en tâche de fond. (priorité -5) Les options d'impression sont accessibles via un panneau de configuration. Par exemple impression en condensé avec comme en-tête le nom du fichier et une numérotation automatique des pages. INDISPENSABLE ! (Pour utilisateurs du Workbench 2.0 seulement) Source C compris.
UIK.LHA : Distribution libre de l'User Interface Kit. Entrez dans le monde de plus en plus convoité de la POO (Programmation Orientée Objet)
UIKShow : Programme de démonstration permettant la visualisation d'une image IFF. Ecrit en C et utilisant l'uik.library et quelques objets externes UIK. A ne pas rater le "look" du sélecteur de fichier. Source C compris.
GLC : Les listes chaînées n'auront plus de secrets pour vous après avoir pris connaissance de GLC. Vous pourrez ainsi récupérer des routines qui les manipulent et explorer un exemple de gestion de ces listes chaînées. Source C compris.
ICONES : Quelques ICONES qui pourront vous servir dans votre propre environnement workbench.
ARexx : Petit utilitaire ARexx qui rajoute "une ligne de commentaire" à chaque saut de page. TRES UTILE. Listing source ARexx compris et facile à comprendre même pour le débutant.
RtPop : Utilisation d'une requête (employant la reqtools.library) en vue de remplacer la commande CLI/SHELL ASK. Réponse par défaut paramétrable et utilisation d'un temps maximum d'attente. Source C compris.
Bourse : (Scrolling part III) Gestion complète du scrolling sans passer par la req.library. L'utilitaire "bourse" illustre cette technique en dessinant le diagramme d'un titre ou d'un CAC40 ou plus généralement de tout ce qui évolue au cours du temps et qui est chiffrable. Source C compris.
ReadPixel : Programme s'adressant aux programmeurs et permettant de prendre connaissance de la position des coordonnées de la souris sur l'écran. Utile pour créer des interfaces personnelles. Source C compris.
Articles : **Manuel ASM suite** : la 4ième partie. On quitte le cadre général pour attaquer la programmation assembleur spécifiquement AMIGA. **ExecRexx** : Article au sujet d'un petit utilitaire permettant de rendre exécutable un source ARexx. Exemple source ARexx compris.**Débutant Amos Part IV** : "les structures de programmation". Quelques mots au sujet d'Amos Pro.

[ ] Je désire recevoir le numéro # 4 du ***Petit Amiga Illustré*** au prix de 40 Frs.
[ ] Je désire m'abonner à 11 numéros à partir du no [__] au prix de 220 Frs.

Nom : ..... Prénom : .....
Adresse : .....
Code : ..... Ville : .....
Ci joint mon règlement par [ ] Chèque, [ ] Mandat - Carte bleue (abonnement uniquement).

Carte Bleue no : !_____!_____!_____!_____! Exp : !_____!_____!

Adressez ce bon à : FDS - FREE DISTRIBUTION SOFTWARE, "Le P.A.I.", Boîte Postale 134, 59453 LYS LEZ LANNOY CEDEX