ECOLE NATIONALE
DES SCIENCES
GEOGRAPHIQUES
CERSIG

Module d'initiation à la programmation MAP-BASIC V.5.0.1

Auteur Pascal Barbier

Ancienne version du document: V 1.1 éditée le: mardi 27 janvier 1998
imprimée le: mardi 27 janvier 1998

Ancienne version du document V 1.2 éditée le: mardi 28 décembre 1999
imprimée le: mercredi 25 janvier 2000

Version du document: V 2.1 éditée le: jeudi 06 décembre 2001
imprimée le: jeudi 06 décembre 2001

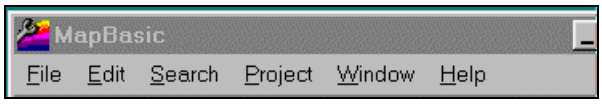
TABLE des MATIÈRES

I- MAP BASIC : l'environnement de développement

Map Basic offre un environnement de développement comprenant un éditeur de texte, pour saisir les programmes, un compilateur pour créer une application exécutable, un éditeur de lien pour créer une application complexe en la décomposant en différents modules « objets », et, enfin une aide en ligne. Malheureusement celle-ci est encore uniquement en anglais. A noter que MapInfo Professionnal est livré avec une documentation Map Basic depuis la Version 6.5
Les programmes exécutables produits par Map Basic ne peuvent que s'exécuter que sur micro ordinateur disposant d'un moteur Map Info. Il existe des relations très fortes entre les versions de MapInfo et les versions de MapBasic utilisées.

1-L'éditeur de texte Map Basic

L'interface de l'éditeur de texte de Map BASIC 5.0.1 se présente de la manière suivante:



File / New Ouvre un nouveau fichier source
File/Open Ouvre un fichier source existant (*.mb)
File/Close Ferme un fichier source Map Basic
File/Close All Ferme tous les fichiers sources Map Basic
File/Save Fait une copie du fichier sources Map Basic courant
File/Save As Fait une copie du fichier sources Map Basic courant en le renommant
File/ Print Imprime un fichier source (après Printer SetUp)

Edit/... Permet d'annuler la dernière opération réalisée, ainsi que les opérations classiques (couper, copier, coller.

Search/ Find Permet de retrouver une chaîne de caractère dans le fichier source.
Search/ Go to Line Permet de se positionner sur une ligne donnée (utile pour le déboguage)

Project/Compile current file Compile le fichier source Map Basic courant (créé le .mbx)
Project/Run Exécute le .mbx. (Cette opération nécessite le moteur MapInfow.exe)
Project/Get Info Permet de savoir le fichier qu'on manipule et la date de sa dernière compilation

Windows Sert à gérer les fenêtres Windows de manière à les organiser en tuile ou en cascade. Ce menu sert également à modifier le corps des caractères utilisés dans l'éditeur

Help Accède à l'aide en ligne. La touche F1 donne accès directement à l'aide du mot sélectionné.

On peut aussi saisir le texte d'un programme Map Basic avec un autre éditeur. Dans ce cas les opérations decompilation sont un peu différentes . Dans ce cas se reporter au guide utilisateur.

2-Déboguage

En cas d'erreur la compilation liste les fautes rencontrées, en indiquant le numéro de ligne ou se trouve le problème. Un double clic sur cette indication positionne le curseur sur la ligne dans l'éditeur MapBasic.
Certaines erreurs sont simples à trouver. pour les plus compliquées d'entre elles, l'instruction **STOP** permet de scruter le contenu des variables du programme dans la fenêtre Map Basic.

ϟ ! le débugueur de MapInfo est assez verbeux et assez imprécis. Une seule faute de syntaxe peut produire plein de lignes d'erreurs, et beaucoup de contrôles qui devraient être fait ne le sont pas. Pour les progammeurs débutants la phase de mise au point du programme entre la compilation et l'exécution régulière n'est pas négligeable.

EXERCICE: Prise en main de l'éditeur - premier lien avec MapInfo
Sous Map Info créer une analyse thématique de la table Adlirois par le champ « Type ».
Sauvegardez le document (workspace)
Ouvrez (.WOR) en .MB
Sous Map Basic, compilez le .mb .
Trouvez les éventuelles erreurs et supprimez les en ajoutant le caractère ['] devant la ligne et exécutez le .mbx sous MapInfo…

II- MAP BASIC La structure minimale de programme

Un programme Map Basic doit être accompagné de commentaires. Un commentaire est précédé par le symbole apostrophe.(')

Pour une meilleure lisibilité du code il est préférable de bien identifier le début et la fin de chaque procédure et de respecter les critères d'indentation des boucles et des tests.

```
'================================================================================
'          Programme écrit en MAP-BASIC V5.1       par  :.....
'                                Unité:
'================================================================================
'Date de création                  : jj mm aa
'Date de dernière modification :jj mm aa
'Nom du Fichier source:  Nom micro/nom du disque/nom des répertoires/nom du fichier
'------------------------------------------------- COMMENTAIRES SUR LE PROGRAMME ----------------------------
' Bla bla bla.. ce qu'il fait, les conditions d'utilisation, les contraintes, les limites etc...
'--------------------------------------------------------------------------------------------------------------------------------

'-------------------------------------------------------------------------Déclaration des procédures
 Declare Sub Main

'=============================================================================MAIN
'------------------------------------------------------ COMMENTAIRES SUR LE MODULE ------------------------------
' Procédure principale  bla bla bla ce qu'elle fait, ce qui rentre, ce qui sort...
'--------------------------------------------------------------------------------------------------------------------------------
 Sub Main

 end sub
'=========================================================================== FIN MAIN
```

Sous Windows 95 la taille maxi d'une fenêtre éditeur est de 64 KO. Elle est de 50 Ko sous Windows 3.1.
Au-delà, il faut découper le programme en plusieurs parties et créer un fichier projet. (PROJECT FILE)

EXERCICE Exo_0 :Prise en main de la compilation
Saisir, compiler et exécuter ce programme.
Gérer les modifications de corps de caractères,
Apprendre à se positionner sur une ligne qui provoque une erreur de compilation. (Par exemple, après avoir fait fonctionner ce programme, modifiez une ligne de commentaire en retirant l'apostrophe initiale.)

III- Eléments d'interface (introduction)

La création de l'interface utilisateur est une composante importante de chaque application. MapBasic fournit les outils nécessaires pour personnaliser l'interface MapInfo. En écrivant un programme MapBasic, on peut créer des menus, des boites de dialogues, gérer des fenêtres et créer des boutons. Nous reviendrons en détail sur les éléments d'interface aux chapitres VIII et IX. Pour l'instant nous allons découvrir quelques outils d'interface, de manière simplement pragmatique, sans aborder les principes qui les régissent.

1-Afficher un message dans une boite bloquante

Note « *message* » +Variable

EXERCICE: Premier pas dans la gestion d'interface.
Ajouter le message « Bonjour » dans le programme précédent. Compiler, puis exécuter.

2-Créer un dialogue qui valide ou non une proposition

Ask (« *proposition* » , « *message_OK* » , « *Message* »)

3-Créer et afficher un menu déroulant dans la barre de menu Map Info

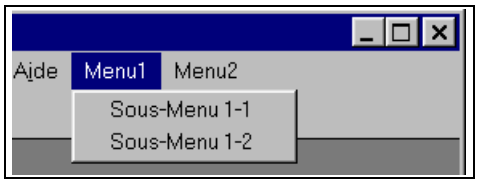
Avec Map Basic il est possible de modifier la barre des menus en ajoutant de nouveau article avec les options suivantes :
Create Menu *NomduMenu* [**ID** *menuId*] **As** *Nomd'unArticledeMenu*[**ID** *ArticleId*]
[**HelpMsg** *Message d'aide*]
{ **Calling** *Nomd'une procédure/* **As** *NomdunMenu hiérarchique*}

Alter Menu Bar {**Add/Remove**} {*NomduMenu/* **ID** *menuId* }

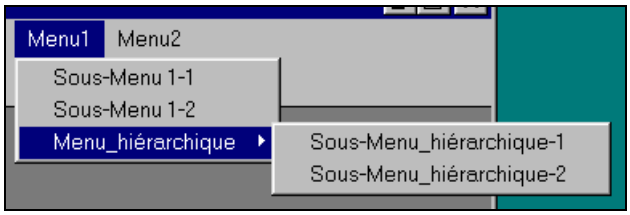
ϟ ! Dans la fenêtre Map Basic pour avoir de l'aide sur une syntaxe, sélectionnez le mot et tapez la touche F1

La création d'un raccourci clavier, apparaissant sous la forme d'une lettre soulignée sous Windows 95, est possible en faisant précéder la lettre retenue d'un &.(ET commercial). La combinaison de touche réunissant la lettre qui suit le & ainsi que la touche Alt active l'option du menu.

EXERCICE 1-1 première partie Savoir modifier la barre de menu, créer des menus, créer des articles :
ajouter dans le module Main deux nouveaux menus (Menu1 et Menu2)
Menu1 donnant Accès à Sous Menu1_1 et Sous Menu1_2 (voir figure)
Menu2 donnant Accès à Sous Menu2_1, Sous Menu2_2 et Sous Menu2_3
Ajouter un message d'aide sur le sous menu 2_2., et compiler puis exécuter.



EXERCICE 1-1 deuxième partie: créer des menus hiérarchiques
ajouter dans le menu Menu1 un troisième article qui ouvre sur un sous menu hiérarchique qui comporte lui aussi deux articles



Sous MAP INFO le fichier MAPINFOW.MNU contient la définition du menu et des boutons « standard » de MAP INFO.

4-Corrigé des exercices sur les Menus déroulants

```
Declare Sub Main
'=========================================================================MAIN
'------------------------- COMMENTAIRES SUR LE MODULE -------------------
' Procédure principale
'-------------------------------------------------------------------------

 Sub Main

     Create Menu "Menu_hiérarchique" As

     "Sous-Menu_hiérarchique-1" ,
     "Sous-Menu_hiérarchique-2"


     Create Menu "Menu1" As

     "Sous-Menu 1-1" ,
     "Sous-Menu 1-2" ,
     "Menu hiérarchique"   As "Menu_hiérarchique"



     Create Menu "Menu2" As

     "Sous-Menu 2-1" ,
     "Sous-Menu 2-2" HelpMsg
          "Petit message d'aide Sous-Menu 2-2"
"Sous-Menu 2-3"

     Alter Menu Bar Add "Menu1"
     Alter Menu Bar Add "Menu2"

 end sub
'============================================================= FIN MAIN
```

IV- Les Types et les variables

1-Les types prédéfinis

On retrouve en Map Basic la plupart des types présents dans les autres langages pour manipuler les entiers, les booléens, les flottants ou les caractères.

Smallint	2 Octets	-32767 à + 32767
Integer	4 Octets	-2 147 483 647 à +2 147 483 647
Logical	1 Octet	False (0) ou True (1)
Float	8 Octets	
String	jusqu'à 32767 caractères	
String * *longueur*	Où *longueur détermine la longueur de la chaîne*	Chaîne de longueur fixe remplie avec des blancs

mais aussi des types spécifiques pour utiliser des tables d'informations ou les graphiques propres à un SIG

Date	4 Octets (2 pour année ; 1 pour mois et 1 pour jour)	MM/JJ/AAAA
Objet	n Octets	Objet graphique
Alias		Nom de colonne
Pen		caractéristiques d'une ligne
Brush		caractéristiques d'un remplissage
Font		caractéristiques d'un texte
Symbol		caractéristique d'un symbole

2-Les types utilisateurs

L'instruction **Type** permet de déclarer les types structurés utiles au programmeur. Les déclarations de type doivent se faire au niveau global. Une variable déclarée sur un type utilisateur ne peut pas être passée par valeur.

Type *NomduType*
    ComposantduType **As** *type_variableduComposant*
    [...]
End Type

ϟ ! On ne peut pas déclarer un type dans la fenêtre MapBasic de Map Info

Déclarer des types structurés permet de limiter le nombre de paramètre passés à un module. Cela améliore la solidité de la programmation en la rendant plus lisible, et en évitant les risques de permutation de deux paramètres du même type.

Ainsi, il est plus astucieux de passer une structure Point comportant un champ X et un champ Y que deux paramètres distincts un X et un Y.

EXERCICE: Traduction d'une classe d'un modèle de données sous Map Basic.
Déclarer un type utilisateur caractérisant un stagiaire, par son nom, son prénom, son âge

3-Les variables

Un type correspond à un potentiel. Une variable correspond à une réalisation d'un potentiel auquel elle est associée. La déclaration d'une variable se fait par l'instruction

Dim *NomVariable* **As** *Type*

Déclarer un tableau nécessite de faire suivre le nom de la variable du nombre d'éléments du tableau entre parenthèse.

Dim *NomVariableTableau (Nb_éléments)* **As** *Type_d'un_élément_du_tableau*

Les variables ont une portée de validité qui est le module dans lequel elles sont déclarées. Elles sont donc inconnues à l'extérieur du module.

Les variables peuvent être déclarées en Global. Elles sont alors connues (et donc modifiables par l'ensemble des procédures). Ces déclarations globales, parfois présentées comme des optimisations, sont à utiliser avec discernement car elles rendent un programme beaucoup moins facile à maintenir en le rendant complexe.

Une variable chaîne de 3 caractères est définie ainsi : **Dim** *NomVariable* **As** *String *3*.

Dans une variable chaîne de longueur N toute affectation d'une chaîne de longueur M inférieure à N sera complétée par des caractères blancs par Map Basic.

4-Les constantes

Map Basic permet de déclarer des constantes. Par exemple: *pi*=3.1415 ou *emploi* = « programmeur »

EXERCICE: Savoir affecter une variable d'un type déclaré par l'utilisateur

Déclarer une variable du type « Tstagiaire » dans le module Main. Affecter le champ nom avec votre nom personnel dans le programme map Basic.

Modifier l'affichage de la boite bloquante, de manière à ce qu'elle souhaite le bonjour à la valeur affectée dans le champ nom

V- Les Modules

Les modules sont: - le programme principal (MAIN)
- les procédures sans type (SUB)
- les procédures avec type (FUNCTION)

Un module est composé d'une partie déclarative et d'une partie descriptive (ou corps).
L'instruction DECLARE permet de déclarer l'entête complète d'un module. Cette déclaration est obligatoire; son absence provoque une faute de compilation.
Un programme comportant une déclaration de procédure (sans le corps) et un appel à cette procédure créera en compilation un .MBO (Map Basic Object). Les .mbo sont liés à une application par des « Project Files ».(compilation séparée, puis édition de lien commune). Le Project File devra obligatoirement lier ce module au module comportant le corps de la procédure, lors de la compilation.

1-La procédure main

C'est la première appelée à l'exécution d'un programme. Elle n'est pas obligatoire, mais rend la lecture d'un programme plus simple.

Declare Sub Main
Sub Main
Liste d'instructions
End Sub

2-Les procédures sans type

L'entête d'une procédure doit être déclarée, puis le corps doit être décrit.

Declare Sub *MaProcedure* **[([ByVal]** *parametre* **As** *var_type* **[, ...])]**

Sub MaProcedure (liste de paramètre formels*)
Liste d'instructions
End Sub

L'appel à la procédure sans type se fait par l'instruction **Call**. *MaProcedure* **(liste de paramètre actuels*)**

*Les paramètres formels (à la déclaration) et actuels (à l'appel)d'une procédure sont typés.

- Ils peuvent être en entrée seulement. C'est à dire que dans ce cas le module utilise le paramètre mais ne le modifie pas. Il doit alors être passé par **valeur** de la manière suivante:
ByVal *NomduParamètre* **As** *Type du Paramètre*

- Ils peuvent être en entrée /sortie. C'est à dire que dans ce cas le module utilise le paramètre et le modifie par affectation. Il est alors passé par **adresse** de la manière suivante:
NomduParamètre **As** *Type du Paramètre*

ϟ ! Le passage par adresse est le mode par défaut de Map Basic

EXERCICE 1-2 première partie:. Savoir appeler une procédure depuis un menu
: déclarer une procédure sans type nommée « Ma_Procedure2_2 ». qui se contente de faire apparaître une boite bloquante indiquant le nom de la procédure. Faire en sorte que cette procédure soit appelée par le sous Menu 2_2 de l'exercice précédant . Noter que l'appel à une procédure sans type se fait par l'utilisation de « Calling » et non pas de « Call » dans un menu. (voir chapitre IX pour approfondissement des appels de procédures dans les objets d'interface.

3-Les procédures avec type

On peut redéclarer les Fonctions Map Basic. Lors d'un appel, ce sera la nouvelle fonction qui sera appelée.

Declare Function MaFonction **(liste de paramètre formels)**
Function MaFonction **(liste de paramètre formels)**
Liste d'instructions
End Function

Les paramètres d'une procédure typée doivent être passés par **valeur**.

EXERCICE 1-2 deuxième partie: Savoir appeler des procédures avec et sans type

Déclarer une procédure avec type nommée « Ma_Fonction ». qui renvoie le carré d'un entier passé en paramètre. Faire en sorte que Ma_Procedure2_2 déclare une variable entière et l'affecte.
Faire afficher le contenu de l'affectation dans une boite bloquante <u>avant et après</u> l'appel à la fonction Ma_Fonction.

<u>4-Corrigé des exercices sur les déclarations de types utilisateurs et les procédures avec type et sans type</u>

```
'================================================================
'            Programme écrit en MAP-BASIC V5.0  par  :Pascal Barbier            :
'================================================================
'------------------------ COMMENTAIRES SUR LE PROGRAMME -------------------------
'Exercice de déclaration de procédure avec et sans type et de leurs appels
'                              et  de déclaration de type utilisateur
'--------------------------------------------------------------------------------
' Déclaration des procédures
'--------------------------------------------------------------------------------
 Declare Sub Main
 Declare Sub Ma_Procedure2_2
 Declare Function Ma_Fonction ( ByVal x As Integer) As Integer
 Type  Tstagiaire
            nom  As String
            prenom As String
            age As Smallint
end type
'================================================Ma_Fonction

'------------------------ COMMENTAIRES SUR LE MODULE -----------------------------
' Function Ma_Function ...
'--------------------------------------------------------------------------------
 Function Ma_Fonction (ByVal x As Integer) As Integer
            Ma_Fonction = X*X
 end Function
'=============================================   Fin MaFonction

'============================================= Ma_Procedure2_2
'------------------------ COMMENTAIRES SUR LE MODULE -----------------------------
' Procédure Ma_Procedure2_2 ...
'--------------------------------------------------------------------------------
 Sub Ma_Procedure2_2

 Dim MaVariable As Integer
            MaVariable =5
'            note "on est dans Ma_Procedure2_2"
            note " valeur avant appel "+ MaVariable
            MaVariable = Ma_Fonction (MaVariable)
            note " valeur après appel "+ MaVariable
end Sub
'============================================= Fin Ma_Procedure2_2

'================================================== MAIN
'------------------------ COMMENTAIRES SUR LE MODULE -----------------------------
' Procédure principale
'--------------------------------------------------------------------------------
Sub Main
   Dim Stagiaire As Tstagiaire
    Stagiaire.Nom ="TOTO"
   note "Bonjour "+Stagiaire.nom
    Create Menu "Menu1" As
    "Sous-Menu 1-1" ,
    "Sous-Menu 1-2"
    Create Menu "Menu2" As
    "Sous-Menu 2-1" ,
    "Sous-Menu 2-2"
            HelpMsg "Petit message d'aide Sous-Menu 2-2"
            Calling Ma_Procedure2_2,
    "Sous-Menu 2-3"
    Alter Menu Bar Add "Menu1"
    Alter Menu Bar Add "Menu2"
end sub
'================================================== FIN MAIN
```

VI- Les Boucles et les Tests

1-La boucle bornée

For *Variable* **=** *Borne Initiale* **to** *Borne Finale* [**Step** *valeur d'incrément*]
    liste d'instruction....
Next

On peut sortir d'une boucle bornée quelle que soit la valeur de la variable de contrôle avec l'instruction **ExitFor**

2-La boucle tant que

While *Condition*
    liste d'instruction....
Wend

La boucle **Tant que** n'est exécutée que si la condition initiale est vraie.

On peut sortir d'une boucle **Tant que**, quelle que soit l'état de la condition de contrôle avec l'instruction **Goto** label

3-La boucle jusqu'à ce que

Do
    liste d'instruction....
Loop Until *Condition*

La boucle **Jusqu'à ce que** est toujours exécutée au moins une fois même si la condition initiale est fausse.

4-La boucle infinie

Do
    liste d'instruction....
Loop

La boucle **Infinie** n'est interrompue que grâce à un débranchement provoqué par un **Goto** ou un **ExitDo**

5-Le test

If *Condition1* **Then**
    liste d'instruction....
[**Elseif** *Condition2* **Then**
    liste d'instruction....]
[**Elseif** *Conditionx* **Then**
    liste d'instruction....]
[**Else** *Condition*
    liste d'instruction....]
End If

6-Le test à choix multiple (CASE)

Do Case *Expression*
    Case Valeur [,Valeur]
    liste d'instruction....
    [**Case**...]
    [**Case Else**
        liste d'instruction....]
End Case

7-Arrêts

End Program Interrompt une application MapBasic
End MapInfo Interrompt une application MapBasic **et** sort de MapInfo

EXERCICE 1-3: Savoir écrire une procédure typée avec paramètres en entrée

Écrire la procédure typée* Max *(de type integer) qui reçoit deux paramètres de type integer et qui renvoie la valeur maximum

```
'=============================================================================
'     Programme écrit en MAP-BASIC V 5.0.1     par  :Pascal Barbier
'
'=============================================================================
'
'Date de création                  : 07 décembre 2001
'Date de dernière modification     : 07 décembre 2001
'
'------------------------- COMMENTAIRES SUR LE PROGRAMME ----------------------
'Exercice de création de procédure typée
'------------------------------------------------------------------------------

'------------------------------------------------------------------------------
' Déclaration des procédures
'------------------------------------------------------------------------------
 Declare Sub Main
 Declare Sub saisie_nombre
 Declare function Max ( ByVal x,y As Integer) As Integer

'============================================================================== Max
 Function Max ( ByVal  x, y As Integer) As Integer
 if x>Y then Max = x
     else max = y
 end if
 end Function
'============================================================================ FIN Max

'=============================================================================
saisie_nombre
 Sub saisie_nombre

 Dim NB1,NB2 As integer

      NB1= 181
      NB2= 23
      Note "Le plus grand des deux nombres est : " + Max ( NB1,NB2)
 end sub
'============================================================== FIN saisie_nombre


'============================================================================== MAIN
'------------------------- COMMENTAIRES SUR LE MODULE -------------------------
' Procédure principale
'------------------------------------------------------------------------------
 Sub Main
     Create Menu "Fonction MAX" As
       "Calcule le Maxi de 2 nombres"
          Calling saisie_nombre

     Alter Menu Bar Add "Fonction MAX"
 end sub
'============================================================================ FIN MAIN
```

VII- Les Tables

Une table MapInfo est un ensemble de fichier qui permettent de gérer des lignes et des colonnes. Chaque ligne représente un « objet » géré par la table. Cet objet est décrit par autant d' « attributs » qu'il y aura de colonnes. une colonne particulière peut contenir un lien vers un fichier qui contiendra réellement la géométrie (points, lignes, polylignes, polygones). Map info peut également gérer des tables qui contiennent des images La structure minimale d'une table MapInfo est donc composées de 3 ou 4 fichiers :

Le .TAB

C'est un fichier éditable qui contient la description de la structure de la table; Par exemple:

```
!table
!version 300          -> version minimale de Map Info capable d'ouvrir cette table !
!charset WindowsLatin1

Definition Table
  Type NATIVE Charset "WindowsLatin1"
  Fields 13
    ID_COM Integer ;
    NOM Char (50) ;
    NUMERO Char (5) ;
    STATUT Char (1) Index 3 ;
    SURFACE Integer ;
    POPU Integer Index 1 ;
    NUM_CAN Char (2) ;
    NUM_ARR Char (1) ;
    NOM_DEP Char (30) ;
    NUM_DEP Char (2) ;
    NOM_REG Char (30) ;
    NUM_REG Char (2) Index 2 ;
    ID Integer ;
```

Le .DAT

C'est un fichier presque éditable (par exemple sous EXCEL) qui contient les information attributaires

Le .ID

C'est un fichier non éditable qui porte la relation entre attributs et géométrie d'un objet.

Le .MAP

C'est un fichier qui contient les informations géométriques (Géométrie des points, ligne, faces) ainsi que le style d'affichage de ces objets géométriques (Couleur, forme..). Si le « ;MAP » est présent, alors le « .TAB » contient un champ Obj, qui permet, pour chaque objet de pointer vers sa géométrie contenue dans le « .MAP »
D'autres fichiers d'index (.IND) et espaces de travail (.WOR) peuvent compléter cette description.

1-Manipuler les tables

Grâce à Map Basic il n'est pas nécessaire de manipuler ces fichiers pour travailler avec l'information car des instructions de « haut niveau » cachent l'accès à ces fichiers et permettent de travailler sur l'information apparente, la table et non pas sur l'information réelle stockée sur disque en fichiers. De nombreuses instructions de manipulations de table existent. Voici les principales.

Ouvrir une table existante

Open Table *nom du fichier* [**As** *Nom de la table*] [**Hide**] [**Readonly**] [**Interactive**]

Fermer une table existante

Close Table *Nom de la table* [**Interactive**]

Créer une nouvelle table

Lors de la création d'une table, on précise les noms des champs. Ceux ci doivent faire moins de 32 caractères. Ils peuvent comporter des lettres des nombres et des underscore[_]. Un nom de champ ne peut pas commencer par un nombre

Create Table *Nom de la table* (*colonne type_de_la_colonne*[, ...])
[**File** *TableSpec*] [{**Type Native/Type DBF**[**Charset** *Set de Caractères*]}]
[**Version** *Version*]

ϟ ! Cette option ne peut pas être utilisée pour créer la colonne obj! (cf fonction Create Map)

Modifier une table existante

Alter Table *Nom de la table* **(**
[**Add** *Nom_de_la_colonne type_de_la_colonne*[, ...]
[**Modify** *Nom_de_la_colonne type_de_la_colonne*[, ...]
[**Drop** *Nom_de_la_colonne* [, ...]]
[**Rename** *Ancien_Nom_de_la_colonne Nouveau_Nom_de_la_colonne*[, ...]]
[**Order** *Nom_de_la_colonne, Nom_de_la_colonne*[, ...]]**)**

Créer et mettre à jour une colonne d'une table existante à partir d'une autre table

Add Column *Nom de la table* **(***colonne*[Type de la colonne])
{Values *const*[,const]/ **From** *Table_Source*
Set to *expression*
[**where {***colonne de destination= colonne source/***whithin/Contains/Intersects}**]
[**dynamic**]**}**

Créer un index sur une table existante

Create Index On *Nom de la table* **(***colonne***)**

Détruire un index sur une table existante

Delete Index *Nom de la table* **(***colonne***)**

Rendre une table existante cartographiable

Create Map For *Nom de la table* **(***colonne***)** [**CoordSys...]**
Si on omet de préciser la clause Coordsys, la table créée sera dans le système terrestre Latitude / longitude. Dans ce cas le **Bound** aussi est fixé par défaut. Il est assez étendu pour couvrir la terre entière. Dans ce cas les coordonnées sont précises au millionième de degré, soit à peu prés 10 centimètres.

Insérer de nouvelles lignes dans une table

Insert into *Nom de la table* [**(***liste colonnes***)] {Values(***liste valeurs***)]**/ **Select** *liste colonnes* **From** *Table***}**

Modifier des lignes dans une table

Update *Nom de la table* **Set** *colonne* **=** *expression*[,*colonne = expression*]
[**Where RowId =** *Rang de la ligne]*

Compacter une table

Pack Table *Nom de la table* [**Graphic/Data**]

!Pack détruit physiquement les enregistrements qui sont préalablement détruits logiquement par un **Delete.** Il faut faire suivre Pack de **Graphic** ou **Data** (**ou les deux).** Par défaut **Pack** ne tasse rien et ne produit pas d'erreur d'exécution. Pour tasser une table elle doit préalablement avoir été sauvegardée par un **Commit**

Sauvegarder une table sur le disque

Commit Table *Nom de la table* [**As** *FileSpec*] [{**Type Native/Type DBF**[**Charset** *Set de Caractères*]}]
[**CoordSys** ...] [**Version** *Version*]

Annuler les changements intervenus sur une table depuis la dernière sauvegarde sur disque

RollBack Table *Nom de la table*

Détruire tout le contenu d'une table existante (la table subsiste mais vide)

Delete From *Nom de la table*

Cette instruction permet aussi de vider uniquement une colonne et peut être contrainte par une clause where. Exemple : Delete Obj from Tron_Route where Rowid = 10, détruit la géométrie de l'objet de la table Tron_route qui est au rang 10

Détruire une table

Drop Table *Nom de la table* !! très puissant à utiliser avec beaucoup de précautions!!

EXERCICE 2 : Savoir ouvrir une table avec Map Basic
Sauvegarder l'exercice EXO_1 en EXO_2.
Nettoyer le fichier texte des scories des exercices précédent.
Renommer « Menu »1 en « Gere_Table » et « Sous-Menu 1-1 » en « Ouvre Table en dur ».
Avec la table des limites communales linéaires BDCarto nommée Adlirois, créer une procédure OpenTableDur, appelée par le sous menu « Ouvre Table en dur ».
Visualiser la table Adlirois par les fonctionnalités MapInfo Fenêtre/Données et Fenêtre/Carte.
Créer une procédure similaire qui permette de choisir la table à ouvrir grâce à la fonction MapBasic FileOpenDlg. (Ainsi que montré dans la copie d'écran ci-dessous)

2-Gérer les systèmes de coordonnées

Les tables Map Info qui sont cartographiables sont associées à un système de représentation de la terre. Celui-ci se défini de manière complète comme un système de coordonnées ainsi qu'un système de représentation plane. La clause COORDSYS permet de définir ces paramètres. Il est à souligner que CoordSys n'est pas une instruction Map Basic mais que Map Basic inclus la clause CoordSys dans de nombreuses instructions par exemple Set Map, SetCoordsys....

Si votre programme Map Basic fonctionne dans un environnement où une table cartographiable a déjà été ouverte, et que vous souhaitez continuer à travailler dans le même système. (par exemple vous avez déjà ouvert une table de BDCarto) vous utiliserez l'instruction:

Set CoordSys Table *NomdeTable*

Toutes les instructions suivantes qui relèveront d'un aspect géométrique seront dans le même système que la table ouverte sous l'alias *NomdeTable.*

Le fichier système MapInfo.prj contient la description de toutes les projections utilisables avec Map Info. Notamment les systèmes de coordonnées français..

ϟ ! A noter la ligne en gras du Lambert 93 qui ne figure pas dans le Mapinfo.prj de Map Info V 5.5 et qui est à ajouter « à la main ».

```
"--- Systèmes Français Méridien de Paris ---"
"Lambert I Carto - Paris\p27581", 3, 1002, 7, 0, 49.5, 48.598522847174,
50.395911631678, 600000, 1200000
"Lambert II Carto - Paris\p27582", 3, 1002, 7, 0, 46.8, 45.898918964419,
47.696014502038, 600000, 2200000
"Lambert III Carto - Paris\p27583", 3, 1002, 7, 0, 44.1, 43.199291275544,
44.996093814511, 600000, 3200000
"Lambert IV Carto - Paris", 3, 1002, 7, 0, 42.165, 41.560387840948,
42.76766346965, 234.358, 4185861.369
"Lambert I Nord - Paris\p27591", 3, 1002, 7, 0, 49.5, 48.598522847174,
50.395911631678, 600000, 200000
"Lambert II Centre - Paris\p27592", 3, 1002, 7, 0, 46.8, 45.898918964419,
47.696014502038, 600000, 200000
"Lambert III Sud - Paris\p27593", 3, 1002, 7, 0, 44.1, 43.199291275544,
44.996093814511, 600000, 200000
"Lambert IV Corse - Paris", 3, 1002, 7, 0, 42.165, 41.560387840948,
42.76766346965, 234.358, 185861.369
"Lambert 93", 3, 999, 0,  0 ,  0 ,  0 ,  7 , 3 , 46.5 ,  44.0, 49.0, 700000,
6600000
```

Cette table traduit la clause CoordSys complète suivante

CoordSys Earth

[**Projection** type,
datum,
unitname
[, origin_longitude]
[, origin_latitude]
[, standard_parallel_1 [, standard_parallel_2]]
[, azimuth]
[, scale_factor]
[, false_easting]
[, false_northing]
[, range]]
[Affine Units unitname, A, B, C, D, E, F]
[Bounds (minx, miny) (maxx, maxy)]

Cela permet de comprendre pourquoi la coordonnée en Y d'un point donné est différente en Lambert I Nord et en Lambert I Carto de 1 000 000. Cela correspond à un Northing différent (origine absolue des Y différente).

3-Corrigé des exercices sur les ouvertures de tables

```
'=========================================================================
'     Programme écrit en MAP-BASIC V 5.0.1     par  :Pascal Barbier
'
'=========================================================================
'Date de création                  : 21 novembre 1997
'Date de dernière modification     : 20 janvier 1998
'
'------------------------- COMMENTAIRES SUR LE PROGRAMME ------------------
'Exercice de manipulation de table ( ouverture sans affichage )
'--------------------------------------------------------------------------

'--------------------------------------------------------------------------
' Déclaration des procédures
'--------------------------------------------------------------------------
 Declare Sub Main
 Declare Sub OpentableDur
 Declare Sub Opentable
'============================================================= OpenTableDur

'------------------------- COMMENTAIRES SUR LE MODULE ---------------------
' Procédure qui sert à ouvrir une table existante en dur ...
'--------------------------------------------------------------------------

 Sub OpenTableDur

 open Table "adlirois" As limites
 end Sub

'============================================================ Fin OpenTableDur
'================================================================ OpenTable

'------------------------- COMMENTAIRES SUR LE MODULE ---------------------
' Procédure qui sert à ouvrir une table avec un dialogue (FileOpenDlg)
'-------------------------------------------------------------------------------
Sub OpenTable Dim NomFic As String
      NomFic = FileOpenDlg("","","TAB","Ouvrir la Table")
      open Table NomFic
end Sub
'================================================================ Fin OpenTable

'================================================================================ MAIN
'------------------------- COMMENTAIRES SUR LE MODULE ----------------------
' Procédure principale
'---------------------------------------------------------------------------
Sub Main
     Create Menu "Gere Table" As
       "Ouvrir Table en dur" HelpMsg
          "Ouvrir une table MapInfo en dur"
```

```
          Calling OpenTableDur,
      "Ouvrir Table " HelpMsg
          "Ouvrir une table MapInfo par menu"
          Calling OpenTable

     Alter Menu Bar Add "Gere Table"
end sub
'============================================================ FIN MAIN
```

4-Travailler avec les tables

Travailler avec les tables nécessite de savoir aller chercher de l'information sur une table, dans une colonne donnée à une ligne donnée..

Connaître le nombre de tables ouvertes

NumTables()

Connaître les descripteurs d'une table

TableInfo(*Nom de la table, Attribut***)**

Nom de la table est le nom ou le numéro d'une table ouverte. Si aucune table est ouverte, l'appel à **TableInfo** génère une erreur.

ϟ ! *Attribut* est un code qui correspond à une valeur déclarée dans MAPBASIC.DEF (fichier à inclure)

Par exemple:
si Attribut vaut TAB_INFO_NAME (ou 1) l'appel à TableInfo() renverra le **nom** de la table
si Attribut vaut TAB_INFO_NCOLS (ou 4) l'appel à TableInfo() renverra le **nombre** de colonnes de la table

Connaître le nombre de colonnes d'une table ouverte

NumCols(*Nom de la table***)** NumCols ne prend pas en compte la colonne portant la géométrie ni celle contenant l'identificateur de rang de ligne (RowId)

Connaître les descripteurs d'une colonne

ColumnInfo(*Nom de la table*, {Nom de colonne/''COLn''},*Attribut***)**

Nom de la table est le nom ou le numéro d'une table ouverte.
Nom de colonne est le nom de la colonne/ COLn fait référence à la colonne qui est en N ième position
Attribut Est un code qui correspond à une valeur déclarée dans MAPBASIC.DEF (fichier à inclure)

Par exemple:
si Attribut vaut COL_INFO_TYPE (ou 3)l'appel à ColumnInfo() renverra un entier qui déterminera le type de la colonne (1 = CHAR; 2 = DECIMAL; 3 = INTEGER; 3 = SMALLINT..... voir MAPBASIC.DEF)

Utiliser les pointeurs de table

EOT(*Nom de la table***)** EOT renvoie un Logical à TRUE si le pointeur courant de la Table est égal au Pointeur sur la Fin de Table

Fetch {First / Last / Next / Prev / Rec n} from *Nom de la table*

Permet de positionner le pointeur de table sur la première, dernière, nième ligne...

La liste des attributs utilisables avec TableInfo est indiquée en Annexe du présent manuel. On peut noter que l'éditeur MapBasic sait éditer les « .DEF »

EXERCICE 3 : Savoir choisir une table parmi les tables ouvertes et un champ parmi les champs de cette table.

Écrire la procédure qui permet de choisir une table parmi les tables ouverte, puis la procédure qui permet de choisir un champ parmi les champ d'une table. Le mécanisme des boites de dialogue est donné sans commentaires.

***Nota*: le choix se fera en utilisant le mécanisme de dialogue intégré à Mapbasic décrit ci dessous**

```
Dialog                          ' *********Début du  Dialogue
        Title "TITRE DU DIALOGUE" ' ---- le titre du dialogue est inscrit dans la chaîne de caractère
   Control ListBox              '-----1er controle
    Title  Liste                ' ----  Liste est une chaîne de caractère chargée avec les articles de la liste déroulante
                                '       séparés par un point virgule « ; »
    Calling Verificateur        '......appel à une procédure vérifiant que l'opérateur a cliqué un article
    ID 1                        '       Numéro du contrôle
    Value 1                     '       précise que par défaut, l'article de rang 1 est sélectionné
    Into VarLoc                 '       le résultat du choix sera affecté dans la variable locale VarLoc
    Position 10, 92 Height 40   '       précise la position en x et Y de la ListBox ainsi que la taille de la liste déroulante

   Control OKButton             '       Appel au contrôle Bouton OK
    ID 2                        '       Numéro du contrôle
    Disable                     '       par défaut inactif

   Control CancelButton         '       Appel au contrôle Bouton Cancel
    ID 3                        '       Numéro du contrôle

   Control StaticText           '       Appel au contrôle affichage d'un texte statique
    Position 5,10
    Width 160
    Title " Baratin"            '       Libellé du texte dans la chaîne de caractères
 ' ********** Fin du Dialogue.... il existe d'autre type de controle....
```

L'accès à une colonne

Syntaxe de référence à une colonne	**Exemple**
NomdeTable.NomdeColonne	*ADLIROIS.TYPE*
NomdeTable.COLn	*ADLIROIS.COL3*
NomdeTable.COL(n)	*ADLIROIS.COL(3)*

Le Type ALIAS

Ce type est à utiliser si le nom, ou le rang de la colonne est inconnu au moment de l'appel. Par exemple votre utilisateur choisit la colonne avec laquelle il souhaite travailler

Map Basic Utilise un type particulier pour manipuler les colonnes le TYPE prédéfini **Alias**. Le type allias est évalué au moment de l'exécution (run time)

La colonne spéciale RowId

La colonne RowId est une colonne spéciale qui porte le rang de chaque ligne. On peut considérer que RowId agit comme une colonne virtuelle utilisable pour des sélections.

La colonne spéciale Obj

La colonne Obj est un nom spécial de colonne qui réfère au fichier des objets graphiques. Cette colonne n'est jamais affichée dans un browser. Si un objet n'est pas associé à un objet graphique, alors pour cet objet le champ est comme vide. (Le pointeur est à NIL)

EXERCICE 4 : Savoir créer et remplir une table par lecture sélective dans une table source

: Écrire la procédure qui permet de créer, à partir de la Table AdLIROIS une nouvelle table nommée Adlimdep. qui ne contient que les enregistrements des arcs des limites départementales (qui correspondent aux enregistrement qui sont de type 3)

5-Corrigé de l'exercice sur les sélections de tables et de colonnes

```
'================================================================================
'       Programme écrit en MAP-BASIC V 5.0.1 par  :Pascal Barbier

'================================================================================
'Date de création                       : 24 novembre 1997
'Date de dernière modification : 21 janvier 1998
'Nom du Fichier source: P200\\C:\_Barbier\cours_mb\programs\exo_3.mb
'------------------------- COMMENTAIRES SUR LE PROGRAMME ----------------------
'Exercice sur le choix d'une table ouverte et le choix d'un champ de cette table
'--------------------------------------------------------------------------------
' Déclaration des procédures
'--------------------------------------------------------------------------------
 Declare Sub Main
 Declare Function GetTable() AS SmallInt
 Declare Sub Verificateur
 Declare sub Choix
 Declare Function GetChamp ( ByVal Table As SmallInt) As smallInt

'========================================================================== MAIN
' Procédure principale
'--------------------------------------------------------------------------------
 Sub Main

     include "..\DEF\MAPBASIC.DEF"
     Create Menu "Gere Table" As
          "Choisir un Champ " HelpMsg
          "Choisir un champ d'une table MapInfo "
           Calling Choix
     Alter Menu Bar Add "Gere Table"
end sub
'======================================================================== FIN MAIN

'========================================================================= GetTable
' Procédure qui sert à choisir une table ouverte
'--------------------------------------------------------------------------------
Function GetTable() AS SmallInt

Dim    i                                  As Smallint
Dim            ChoixT,
               Table_Choisie,
               Tabeau_Table (10)     As String

       '--------------------- Vérification qu'il y a une table ouverte

       If NumTables()=0 Then Note " Une table doit être ouverte !"
       End If

  '----------------- Sélection des tables ouvertes
     for i =1 to NumTables()
            Tabeau_Table (i) = TableInfo (I,TAB_INFO_NAME)
     Next
     choixT =""
     for i =1 to NumTables()
            ChoixT = ChoixT+Tabeau_Table (i) +";"
     Next

   Dialog ' ********* Dialogue de choix de table
       Title "Choix de la TABLE"
     Control ListBox '-----1er controle
       Title  ChoixT
       Calling Verificateur
       ID 1
       Value 1
       Into Table_Choisie
       Position 10, 92 Height 40

     Control OKButton
       ID 2
       Disable

     Control CancelButton
       ID 3

     Control StaticText
       Position 5,10
       Width 160
       Title " Choississez une table:"
   ' ************ Fin du Dialogue de choix de table
       GetTable = Table_Choisie
end Function
'============================================================== Fin GetTable
'============================================================== GetChamp
' Procédure qui sert à sélectionner un nom de champ
'--------------------------------------------------------------------------------
 Function GetChamp (ByVal Table As SmallInt) As SmallInt

 Dim TabChamp (50)    As String
 Dim i                       As Integer
 Dim ChoixCh          As String
```

```
 Dim Champ                   As SmallInt

    '----------------- Sélection des Champs de la table
      for i =1 to NumCols(Table)
             TabChamp (i) = columnInfo (Table,"Col"+i,COL_INFO_NAME)
      Next
    choixCh =""
    for i =1 to NumCols(Table)
       ChoixCh = ChoixCh+TabChamp (i) +";"
    Next

   Dialog ' ********* Dialogue de choix du champ
       Title "Choix du CHAMP"
     Control ListBox '-----1er controle
       Title  ChoixCh
       Calling Verificateur
       ID 1
       Into Champ
       Position 10, 92 Height 40

     Control OKButton
       ID 2
       Disable

     Control CancelButton
       ID 3

     Control StaticText
       Position 5,10
       Width 160
       Title " Choississez le Champ:"

   ' ************ Fin du Dialogue de choix du Champ
   GetChamp= Champ

 end Function
'============================================================ Fin GetChamp

'=================================================================== Choix
' Procédure qui sert à appeler GetTable et GetChamp et à vérifier ce que renvoient les procédures
'-------------------------------------------------------------------------------
Sub Choix

    Dim NomTable,
       NomChamp       As String
       Dim Choix1             As SmallInt
       Dim Choix2             As SmallInt

              Choix1 = GetTable ()
        Choix2 = GetChamp (Choix1)
        NomTable = TableInfo (Choix1,TAB_INFO_NAME)
        NomChamp = ColumnInfo (NomTable,"COL"+str$(Choix2),COL_INFO_NAME)
              NomTable =NomTable+NomChamp
        note "le champ choisi est " +str$( Nomchamp)

 End Sub

'============================================================== Fin Choix

'================================================== VERIFICATEUR
Sub Verificateur

    If ReadControlValue (1) then
         Alter Control 2 Enable
    else
         Alter Control 2 Disable
    End If
end sub
'================================================ FIN VERIFICATEUR
```

6-Corrigé de l'exercice sur la création de la table sémantique des limites départementales

```
'=========================================================
'            Programme écrit en MAP-BASIC V5.0.1          par  :Pascal Barbier
'=========================================================
'Date de création                          : 25 novembre 1997
'Date de dernière modification :           21 janvier 1998
'Nom du Fichier source: P200\\C:\_Barbier\cours_mb\programsexo_4.mb
'------------------------ COMMENTAIRES SUR LE PROGRAMME --------------------------
'Exercice de travail sur les tables. Savoir recopier le contenu sémantique d'une table à partir
' d'un test sur une valeur dans une colonne de la table de départ
'-------------------------------------------------------------------------------Déclaration des procédures
 Declare Sub Main
 Declare Function GetTable() AS SmallInt
 Declare Sub Verificateur
 Declare sub CreTabDep

'========================================================== MAIN
'------------------------ COMMENTAIRES SUR LE MODULE -----------------------------
' Procédure principale

Sub Main
include "..\DEF\MAPBASIC.DEF"
    Create Menu "Gere Table" As

     "extraire les limites dép. "
       HelpMsg
        "Créer une table contenant les limites départementales "
         Calling CreTabDep
   Alter Menu Bar Add "Gere Table"
end sub
'========================================================== FIN MAIN
'========================================================== CreTabDep
'------------------------ COMMENTAIRES SUR LE MODULE -----------------------------
' Procédure qui sert à créer la table des limites départementales

 Sub CreTabDep
   Dim NomTable              As String
   Dim Table                 As SmallInt
   Dim TabChamp (50)         As String
   Dim i                     As Integer
   Dim Champ                 As SmallInt

   Dim Col_ID_Lim,Col_Precis,
          Col_Type           As Alias
   Dim Val_Id_Lim            As Integer
   Dim Val_Precis,Val_Type   As String
          Table = GetTable ()            ' --- sélection de la table ADLIROIS par menu graphique
          NomTable= TableInfo (table,TAB_INFO_NAME)
   '----------------------------- recherche des champs de la table sélectionnée
          for i =1 to NumCols(Table)
                TabChamp (i) = columnInfo (Table,"Col"+i,COL_INFO_NAME)
          Next
          '----- création de la table destination avec la même structure que la table ADlirois

          create Table Adlimdep ( ID_LIM Integer,PRECIS char(1), TYPE char(1) )

          Col_ID_Lim = NomTable+".ID_LIM"
          Col_Type = NomTable+".TYPE"
          Col_PRECIS = NomTable+".PRECIS"

          fetch first from NomTable
          while not eot(NomTable)
                Val_type =Col_Type
                If Val_Type = "3" then
                      Val_ID_Lim = Col_Id_Lim
                      Val_PRECIS = Col_PRECIS
                      insert into AdLimDep(ID_LIM,PRECIS,TYPE) Values(Val_ID_Lim, Val_PRECIS, Val_Type )
                end if
                fetch next from NomTable
          wend
End Sub
'================================================== Fin CreTabDep
Function GetTable() AS SmallInt        ‘ Voir code de la fonction GetTable dans exercice EXO_3
Sub Verificateur                       ‘ Voir code de la fonction Verificateur dans exercice EXO_3
```

VIII- Les Fenêtres

1-Gérer les fenêtres

Ouvrir une fenêtre

Open Window *nom du type de la fenêtre*

Ouvre une nouvelle fenêtre d'un des types suivants: MapBasic, Satistics, Legend, Info, Ruler, Help, Message

Fermer une fenêtre

Close Windows *Nom du type de la Fenêtre*

Modifier l'état d'une fenêtre

Set Window *Identificateur de la Fenêtre*
[**Position(x,y)** [**Units** *Unité de papier*]]
[**Width x** *largeur* [**Units** *Unité de papier*]]
[**Height x** *Hauteur* [**Units** *Unité de papier*]]
[**Font ..**]
[**Min/Max/Restore**]
[**Front**]
[**Title {** *Nouveau Titre/* **Default}**]
[**Scrollbars {On/Off}**]

Ouvrir une fenêtre Carte associée à une table

Map From *Table* [**,table..]**
[**Position x** [**Units** *Unité de papier*]]
[**Width x** *largeur* [**Units** *Unité de papier*]]
[**Height(x,y)** *Hauteur* [**Units** *Unité de papier*]]
[**Min/Max**]

Ouvrir une fenêtre Browser associée à une table

Browse *Sélection* **From** *Table* [**,table..]**
[**Position(x,y)** [**Units** *Unité de papier*]]
[**Width x** *largeur* [**Units** *Unité de papier*]]
[**Height x** *Hauteur* [**Units** *Unité de papier*]]
[**Row** n
[**Column** n
[**Min/Max**]

La sélection peut être soit un astérisque soit une liste de nom de colonnes séparée par une virgule

Modifier l'état d'une fenêtre carte

Set Map [**Window** *Identificateur de fenêtre*]
[**Center** (*longitude, latitude)*]
[**Zoom..**]
[**Order ..**]
[**Coordsys...**]
. . . . voir help !

Connaître le nombre de fenêtres ouvertes

NumWindow()

Renvoie un SmallInt correspondant au nombre de fenêtres ouvertes.

Connaître le l'identificateur de la fenêtre active

FrontWindow()

Renvoie un SmallInt correspondant au nombre de fenêtres ouvertes.

Obtenir des informations sur une fenêtre

WindowInfo (*Identificateur de fenêtre, attribut* **)**

2-Manipuler les couches graphiques dans les fenêtres Windows

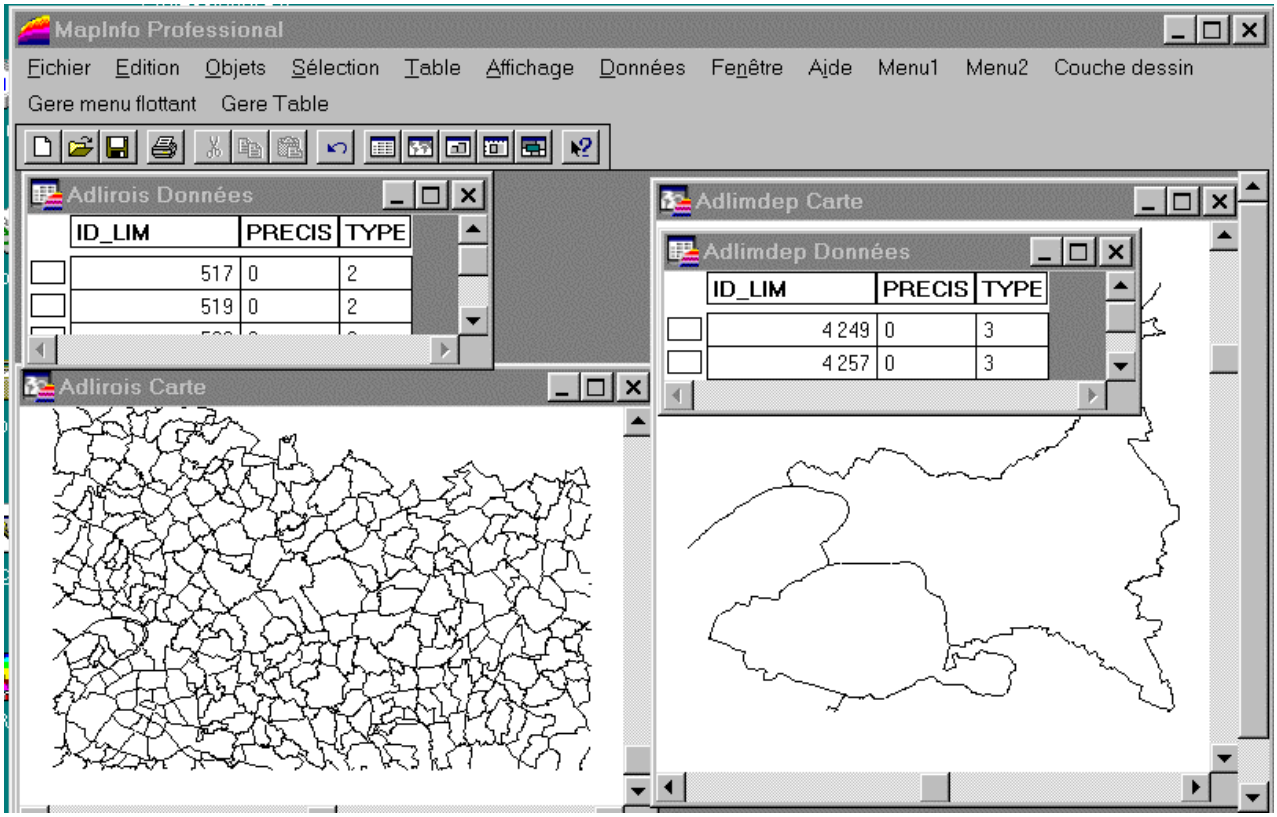
Ajouter une couche à une fenêtre Carte

Add Map [**Window.** *Identificateur de fenêtre***]**
[**Auto**] **Layer** *Table*[*, Table*]

Retirer une couche à une fenêtre Carte

Remove Map [**Window.** *Identificateur de fenêtre***]**
Layer *Numéro de couche*[*,Numéro de couche*]

EXERCICE 5 : Apprendre à transporter la géométrie d'une table et gérer l'affichage dans une fenêtre définie.
A partir de l'exercice précédent, écrire la procédure qui permet de créer, à partir de la Table AdLIROIS une nouvelle table qui ne contient que les enregistrements des arcs des limites départementales (celles qui sont de type 3) et qui transporte la géométrie des arcs.
Faire en sorte que à la création de la nouvelle table, le contenu sémantique s'affiche, ainsi que le dessin des limites départementales dans une fenêtre de taille 10 * 10 cm située à 5 cm en X comme en Y du coin Haut-Gauche de la fenêtre Map Info.
Nota la clause Set Coord Sys sera la suivante: CoordSys Earth Projection 12,12,"m",0.



3-Corrigé de l'exercice sur la création de la table sémantique et géométrique des limites départementales

```
'============================================================
'               Programme écrit en MAP-BASIC V 5.0.1        par  :Pascal Barbier
'============================================================
'Date de création                                :           25 novembre 1997
'Date de dernière modification :                             21 janvier 1998
'
'Nom du Fichier source: P200\\C:\_Barbier\cours_mb\programs\exo_5.mb
'------------------------- COMMENTAIRES SUR LE PROGRAMME ----------------------------
'Exercice de travail sur les tables et les champs graphique. Compléter l'exercice Exo_4 en transportant la géométrie
de la table ADLIROIS et en l'affichant le résultat graphique dans une fenêtre de 10*10 cm
'-------------------------------------------------------------------------------
' Déclaration des procédures
'-------------------------------------------------------------------------------
 Declare Sub Main
 Declare Function GetTable() AS SmallInt
 Declare Sub Verificateur
 Declare sub CreTabDepGeo
'========================================================== MAIN
Sub Main

    include "..\DEF\MAPBASIC.DEF"
    Create Menu "Gere Table" As
"extraire limites géométriques dép."
       HelpMsg
         "afficher les limites départementales dans une fenêtre 10*10 cm"
         Calling CreTabDepGeo
Alter Menu Bar Add "Gere Table"
end sub
'========================================================= FIN MAIN

'=====================================================CreTabDepGeo
'------------------------------------------------ COMMENTAIRES SUR LE MODULE ---------------------------------
' Procédure qui sert à créer la table des limites départementales avec la géométrie des arcs

 Sub CreTabDepGeo

Dim         NomTable  As String
Dim         Table                   As SmallInt
Dim          TabChamp (50)          As String
Dim          i                      As Integer
Dim         Champ                   As SmallInt
Dim         Col_ID_Lim, Col_Precis,
            Col_Type, Col_Geo As Alias
Dim         Val_Id_Lim As Integer
Dim          Val_Precis,
            Val_Type                As String
Dim         Val_Geo                 As Object
            Table = GetTable ()     ' --- sélection de la table ADLIROIS par menu graphique
            NomTable= TableInfo (table,TAB_INFO_NAME)
   '---------------------------- recherche des champs de la table sélectionnée
            for i =1 to NumCols(Table)
                        TabChamp (i) = columnInfo (Table,"Col"+i,COL_INFO_NAME)
            Next
            '----création de la table destination avec la même structure que la table ADlirois

            create Table Adlimdep ( ID_LIM Integer,PRECIS char(1), TYPE char(1) )
            '-------------------- création du champ "obj".... en réalité le fichier .MAP
            Create Map For Adlimdep   CoordSys Earth Projection 12,12,"m",0.

            Col_ID_Lim = NomTable+".ID_LIM"
            Col_Type   = NomTable+".TYPE"
            Col_PRECIS = NomTable+".PRECIS"
            Col_Geo    = NomTable+".Obj"

            fetch first from NomTable
            while not eot(NomTable)
                        Val_type =Col_Type
                        If Val_Type = "3" then
                                    Val_ID_Lim = Col_Id_Lim
                                    Val_PRECIS = Col_PRECIS
                                    val_Geo    = Col_Geo
                                    insert into AdLimDep(ID_LIM,PRECIS,TYPE,Obj)
                                    Values(Val_ID_Lim, Val_PRECIS, Val_Type, Val_Geo )
                        end if
                        fetch next from NomTable
            wend

            Map from AdlimDep Position (5,5) Units "cm" Width 10 Units "cm" Height 10 Units "cm"
            Set Map Zoom Entire

Browse * from Adlimdep
 End Sub
'================================================ Fin CreTabDepGeo
```

4-Questionner les fenêtres et les couches graphiques

Obtenir des informations sur une fenêtre Carte

MapperInfo(*Identificateur de fenêtre, Attribut***)**

Attribut peut prendre différentes valeurs, par exemple

MAPPER_INFO_LAYER : renvoie le nombre de couches de la fenêtre carte (exceptée la couche dessin appelée « Cosmetic Layer » pour Map Basic)

MAPPER_INFO_SCALE : renvoie l'échelle active exprimée dans le rapport (distance terrain/ unité papier)

MAPPER_INFO_CENTERX : renvoie la valeur du X au centre de la fenêtre (Respectivement Y)

MAPPER_INFO_MAXX : renvoie la valeur du Maxi en X affichée dans la fenêtre (Respectivement Y)

etc... voir help.

Obtenir des informations sur une couche affichée dans une fenêtre Carte

LayerInfo(*Identificateur de fenêtre, Numéro de Couche,Attribut***)**

Attribut peut prendre différentes valeurs, par exemple

LAYER _INFO_LAYER : renvoie le nombre de couches de la fenêtre carte (exceptée la couche dessin appelée « Cosmetic Layer » pour Map Basic)

LAYER _INFO_SCALE : renvoie l'échelle active exprimée dans le rapport (distance terrain/ unité papier)

LAYER _INFO_CENTERX : renvoie la valeur du X au centre de la fenêtre (Respectivement Y)

LAYER _INFO_MAXX : renvoie la valeur du Maxi en X affichée dans la fenêtre (Respectivement Y)

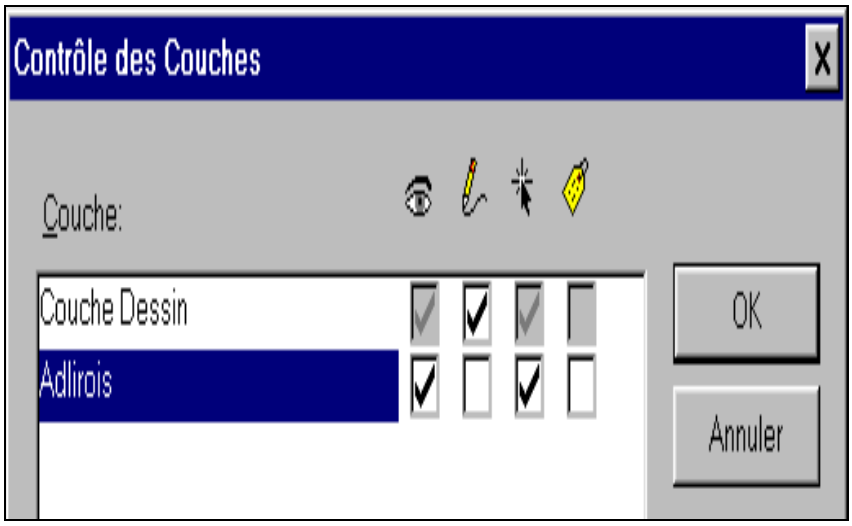
etc... voir help.

Modifier l'état d'une fenêtre carte

Set Map [**Window** *Identificateur de fenêtre*]
[**Center** (*longitude, latitude)*]
[**Zoom..**]
[**Coordsys...**]
. . . . voir help ! Cette fonctionnalité est très puissante. Elle permet de contrôler tous les paramètres d'une fenêtre carte et des couches qui la compose.

EXERCICE 6: Savoir modifier les caractéristiques de la couche dessin

Écrire un programme qui active/désactive à chaque appel la couche dessin, ou qui désactive une couche dessinable en rendant la couche dessin dessinable. L'effet de la fonctionnalité est montré par les captures d'écran suivantes qui correspondent à deux appels successifs à la fonction



5-Corrigé de l'exercice sur gestion de la couche dessinable

'==========

```
'Programme écrit en MAP-BASIC V3.0  par  :Pascal Barbier
'================================================
'Date de création                        : 28 novembre 1997
'Date de dernière modification           : 28 novembre 1997
'
'------------------------ COMMENTAIRES SUR LE PROGRAMME --------------------------
'Exercice sur l'activation de la couche graphique dessinable/non dessinable
'---------------------------------------------------------------------------

'---------------------------------------------------------------------------
' Déclaration des procédures
'---------------------------------------------------------------------------
 Declare Sub Main
 Declare Sub ChangeCosmetic

'================================================MAIN
'------------------------ COMMENTAIRES SUR LE MODULE -----------------------------
' Procédure principale
'---------------------------------------------------------------------------------------------
 Sub Main
 Dim Table          As String

    include "..\MAPBASIC.DEF"

    Create Menu "Couche dessin" As

     "modifier la couche dessin"
       HelpMsg
         "modifier la possibilité de dessiner dans la couche dessin"
          Calling ChangeCosmetic
     Alter Menu Bar Add "Couche dessin"

 end sub
'================================================FIN MAIN

'=========================================CHANGECOSMETIC
'------------------------ COMMENTAIRES SUR LE MODULE -----------------------------
' Procédure qui ouvre une fenêtre carte
'---------------------------------------------------------------------------------------------
Sub CHANGECOSMETIC

Dim     Id_Fen,
        Id_Layer,
        NbLayer         As Integer
Dim i,
        Dessinable      As SmallInt

Id_fen = FrontWindow()              ' --on récupère le numéro de la fenêtre sélectionnée

 Dessinable = MapperInfo(Id_Fen,MAPPER_INFO_EDIT_LAYER)

Do Case Dessinable

        Case -1                                 '---------- Aucune couche editable
                Set Map Layer 0 Editable ON
        Case 0                                  '---------- La couche dessin est dessinable
                Set Map Layer 0 Editable OFF
        Case Else                                       ' --------- il y a une autre couche de dessinable
                NbLayer = MapperInfo(Id_Fen,MAPPER_INFO_LAYERS)
                for i= 1 to NbLayer
                        If LayerInfo (Id_Fen,i,LAYER_INFO_EDITABLE) then
                                                ' ---------- on a trouvé la couche dessinable
                                Id_Layer = i
                                Goto Sortie
                        End if
                next
                sortie:
                Set Map Layer Id_Layer Editable OFF
                Set Map Layer 0 Editable ON

End Case
End Sub
========================================== Fin CHANGECOSMETIC
```

IX- Eléments d'interface (suite)

Map basic utilise le modèle de programmation appelé programmation événementielle.
Un événement est une action que l'utilisateur réalise avec l'interface Homme-Machine (IHM) en utilisant la souris. Ainsi chaque clic souris génère un événement. Par exemple choisir un article dans un menu génère un événement de choix-menu, fermer une fenêtre génère un événement de fermeture de fenêtre.
Quand l'utilisateur génère un événement, le programme doit réagir en conséquence. Il prend en charge la gestion événementielle (to handle en anglais). La procédure concernée agit donc comme un **handler**.

ϟ ! Un handler est une procédure SANS PARAMÈTRE.

Dans un menu déroulant
Ainsi, créer un menu personnalisé est un processus à deux étapes:

Personnaliser la structure du menu Map INFO en utilisant les instruction Create Menu ou Alter Menu

Spécifier un Handler pour chaque article personnalisé, afin de définir ce que le programme doit faire si cet article est choisi. En complément du choix d'une procédure comme handler d'un article de menu, on peut choisir une commande standard MAP INFO.
La liste des commandes standard est contenue dans le fichier **menu.def.** Ainsi un menu personnalisé peut appeler la commande Créer une carte thématique en faisant suivre l'article du menu par le handler suivant:

Calling M_MAP_THEMATIC

Quand une procédure agit comme un handler l'appel est réalisé en utilisant Calling au lieu de l'instruction Call.

Dans un menu type boîte de boutons
Avec les boutons à enfoncer (PushButton), MapInfo appelle le handler au moment ou l'utilisateur clique sur le bouton.
Avec les boutons outils (ToolButton) MapInfo appelle le handler si l'utilisateur choisi l'outils.

1-Les boites de dialogues

Nous avons déjà vu et utilisé des boites de dialogue (Note, Ask, FileOpenDlg( ). Nous allons compléter notre connaissance des boites de dialogue personnalisable et des boutons.

Création d'un dialogue

Toute chose apparaissant dans un dialogue est appelée contrôle. (voir page 105 du User's Guide)

Dialog
[Title *titre* **]**
[Width *w* **] [Height** *h* **] [Position** *x , y* **]**
[Calling *handler* **]**
Control *control_clause* **. . .**

[Control *control_clause* **. . .]**

Le handler de dialogue est la procédure appelée avant que l'utilisateur ne soit autorisé à utiliser le dialogue. C'est très utile pour contrôler l'instruction Alter Control qui active ou désactive les boutons.
Quand l'utilisateur quitte le dialogue, Map Info exécute l'instruction qui suit. Il est alors possible d'appeler la fonction CommandInfo() pour savoir si l'utilisateur est sorti en cliquant sur OK ou bien sur Annule.

La taille et la position de chaque controle est paramétrable. Chaque Control Clause peut être l'un des types suivant:

- Button / OKButton / CancelButton : Bouton avec titre (ou boutons spéciaux OK et Cancel)
- CheckBox : Case à cocher
- GroupBox : Rectangle avec un titre en haut à gauche
- RadioGroup : Ensemble de bouton radio (1 choix parmi n ,à la fois)
- EditText : Boite de saisie
- StaticText : Controle non interactif, pour afficher un texte
- PenPicker /BrushPicker / SymbolPicker / FontPicker : Control Map Basic permettant de gérer les styles
- ListBox / MultiListBox : Liste déroulante émettant 1 choix/ plusieurs choix
- PopupMenu : Apparaît comme un texte avec une flèche, qui permet de faire apparaître une liste pour sélectionner
-

2-Les menus flottant à boutons (Buttons Pad)

Un menu à boutons est une ensemble de boutons 3D dans une fenêtre que l'utilisateur peut déplacer et modifier. Cet ensemble apparaît comme une boite à outils. Les boutons sont de trois types les ToggleButton, qui fonctionnent en boucle, les PushButtons qui sont les boutons à enfoncer et les ToolButton, qui sont des boutons qui servent à activer un outil. Ces trois types d'usages sont très fréquents dans l'utilisation classique de l'interface Map Info.

Création d'un menu flottant de boutons

Create Button Pad {*nom/***ID** *Identificateur de menu flottant* **} As**
type de bouton
[**Title** *Titre du menu flottant*]]
[**Width** *w*]
[**Position(***x,y***)** [**Units** *Nom de l'unité de mesure*]]
[**{Show/Hide}**]

Le *type de bouton* peut être soit un séparateur (Separator) soit une des forme de bouton suivantes:
Push Button, ToolButton,ToggleButton.. La syntaxe associée est la suivante:

[**{PushButton/ToggleButton/ToolButton}**
Calling *Handler*
[**ID** *Identificateur du bouton*]
[**Icon** *n* [**File** *File spec*]]
[**Cursor** *n* [**File** *File spec*]]
[**DrawMode** *Dm_code*]
[**HelpMsg** *Message*]
[**ModifierKeys{on/Off}**]
....

Les icônes disponibles dans MAP BASIC sont listées dans le fichier ICONS.DEF. chaque icône est associée à un numéro icône. L'exécutable ICONDEMO.MBX permet de visualiser ces icônes et à en connaître la valeur d'identifiant. Créer sa propre icône est possible. Mais Map Basic ne fournit pas l'éditeur de ressource nécessaire.(voir p 246 du manuel utilisateur). Cursor permet de définir la forme du curseur après avoir cliqué le bouton et Drawmode permet de définir la manière de travailler sur la fenêtre carte. Cela est très utile pour le boutons de type ToolButton.

Modification d'un menu flottant de boutons

Alter Button Pad ....

3-Les Contrôles de style Map Basic

Map Basic offre des outils appelables depuis un module Map Basic pour modifier les styles des objet géométriques ponctuels, linéaires et surfacique ainsi que des textes. Ce sont des sous parties d'un Dialogue.

Modification d'un style ponctuel

Control SymbolPicker {*nom/***ID** *Identificateur de menu flottant* **} As**
[**Position** *x,y*]
[**Width** *w*]
[**Heigth** *h*]
[**Calling** *Handler*]
[**Value** *Variable initiale*]
[**Into** *Variable de style symbol*]
[**Disable**]
[**Hide**]

Modification d'un style linéaire

Control PenPicker {*nom/***ID** *Identificateur de menu flottant* **} As ....**

Modification d'un style surfacique

Control BrushPicker {*nom/***ID** *Identificateur de menu flottant* **} As ....**

Modification d'un style texte

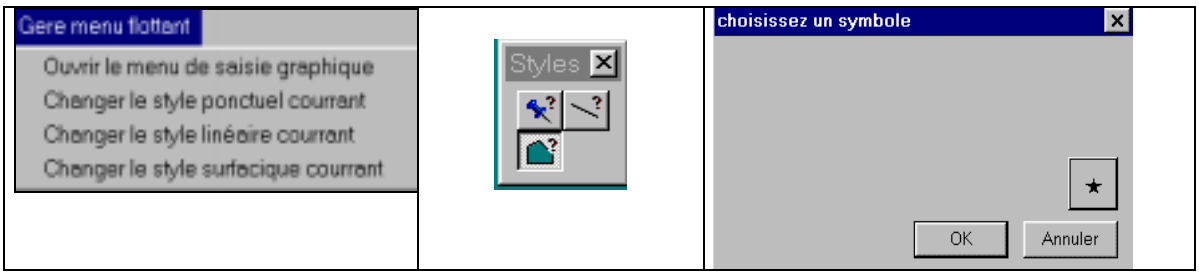
Control TextPicker {*nom/***ID** *Identificateur de menu flottant* **} As...**

EXERCICE 7-1: Savoir gérer les menus flottants

Écrire un programme qui, par menu déroulant génère un Button Pad qui possède un bouton de chaque type (*PushButton/ToggleButton/ToolButton*). le Handler appelé ne fait qu'annoncer la procédure handler par un « Note... »

EXERCICE 7-2: Savoir appeler un handler de deux manières (menu déroulant ou boutons)

Écrire un programme qui, par menu déroulant (figure de gauche) créé un menu flottant (figure du centre) qui gère les styles ponctuels, linéaire et surfaciques grâce aux Pickers Map Basic (figure de droite).



4-Corrigé des exercices sur les menus flottants , les boutons et les contrôles de styles

```
'====================================================================
'              Programme écrit en MAP-BASIC V 5.0.1              par  :Pascal Barbier
'====================================================================
'Date de création                                  : 26 janvier 1998
'Date de dernière modification              : 26 janvier 1998'
'Nom du Fichier source: P200\\C:\_Barbier\cours_mb\programs\exo_7_1.mb
'------------------------- COMMENTAIRES SUR LE PROGRAMME ----------------------
'Exercice sur les menus flottants
'------------------------------------------------------------------------------
' Déclaration des procédures
              Declare Sub Main
              Declare Sub MenuSaisie
              Declare Sub Bye
              Declare Sub ActionToolButton
              Declare Sub ActionPushButton
              Declare Sub ActionToggleButton

'============================================================== MAIN
'------------------------- COMMENTAIRES SUR LE MODULE -------------------------
' Procédure principale
Sub Main
 Dim Table            As String

              include "..\DEF\MAPBASIC.DEF"
              include "..\DEF\ICONS.DEF"
    Create Menu "Gere menu flottant" As
     "Ouvrir le menu flottant"
       HelpMsg          "ouvrir des exemples de menus flottants"
          Calling MenuSaisie,
     "Supprimer Gere Menu Flottant"  Calling Bye
     Alter Menu Bar Add "Gere menu Flottant"
 end sub
'============================================================ FIN MAIN
'============================================================== Bye
'------------------------- COMMENTAIRES SUR LE MODULE -------------------------
' Retire un menu déroulant de la barre de menus
Sub Bye
     Alter Menu Bar remove "Gere menu Flottant"
End Sub
'============================================================ FIN Bye
'================================================== ActionToolButton
'------------------------- COMMENTAIRES SUR LE MODULE -------------------------
'Handler du ToolButton
Sub ActionToolButton
   note "on est dans Action TOOLBUTTON"
End Sub
'================================================ FIN ActionToolButton
'================ ActionPushButton IDEM ACTIONTOOLBUTTON===============
'================ ActionToggleButton IDEM ACTIONTOOLBUTTON ==============
'================================================================== MenuSaisie
'------------------------- COMMENTAIRES SUR LE MODULE -------------------------
' Créée un panneau de boutons de saisie
Sub MenuSaisie

 create ButtonPad  "Menu-Boutons" AS
              ToolButton
                        Icon MI_ICON_LETTERS_T
                        HelpMsg "Changer le style ponctuel"
                        Calling ActionToolButton
              Separator
              PushButton
                        Icon   MI_ICON_LETTERS_P
                        Calling ActionPushButton
                        HelpMsg "Baratin sur ce que fait le bouton à enfoncer"
              Separator
              ToggleButton
                        Icon   MI_ICON_LETTERS_Z
                        Calling ActionToggleButton
End Sub
'================================================ FIN MenuSaisie
```

Deuxième exercice….

```
'=================================================
'          Programme écrit en MAP-BASIC V5.0.1          par  :Pascal Barbier
'=================================================
'Date de création              28 novembre 1997
'Date de dernière modification :          28 novembre 1997
'
'Nom du Fichier source: P200\\C:\_Barbier\cours_mb\programs\exo_7-2.mb
'------------------------- COMMENTAIRES SUR LE PROGRAMME ----------------------------
'Exercice sur les menus flottants
'-------------------------------------------------------------------------------'-----------------------------------------------
' Déclaration des procédures
'-------------------------------------------------------------------------------
 Declare Sub Main
 Declare Sub MenuSaisie
 Declare Sub GereSymbPonct
 Declare Sub GereSymbPen
 Declare Sub GereSymbZone
' ------------------------------------------------------------ choix de déclarer des contrôleurs de symboles globaux
 Global Symbole As Symbol
 Global Ligne       As Pen
 Global Motif       As Brush

'================================================= MAIN
' Procédure principale
'-------------------------------------------------------------------------------
 Sub Main

 Dim Table          As String

    include "..\DEF\MAPBASIC.DEF"
    include "..\DEF\ICONS.DEF"
        Symbole = CurrentSymbol()
        Ligne       = CurrentPen()
        Motif       = CurrentBrush()
    Create Menu "Gere menu flottant" As

     "Ouvrir le menu de saisie graphique"
       HelpMsg      "ouvrir l'outil de saisie graphique"
         Calling MenuSaisie,

     "Changer le style ponctuel courrant"
       HelpMsg      "gérer les symboles ponctuels"
         Calling GereSymbPonct  ,

"Changer le style linéaire courrant"
       HelpMsg      "gérer les symboles linéaires"
         Calling GereSymbPen,

     "Changer le style surfacique courrant"
       HelpMsg       "gérer les symboles surfacique"
         Calling GereSymbZone

     Alter Menu Bar Add "Gere menu Flottant"
end sub
'=================================================FIN MAIN

'=================================================GERESYMBPONCT
'------------------------- COMMENTAIRES SUR LE MODULE ------------------------------
' Procédure qui gère la symbolisation ponctuelle
'-------------------------------------------------------------------------------
 Sub GereSymbPonct

   Dialog
   title "choisissez un symbole"
   Control SymbolPicker
           Position 142,45
           Into Symbole
     ID 1

   Control OKButton
    ID 2

   Control CancelButton
    ID 3
        Set Style Symbol Symbole
end sub
'========================================== FIN GERESYMBPONCT

'=================================================GERESYMBPEN
'------------------------- COMMENTAIRES SUR LE MODULE ------------------------------
' Procédure qui gère la symbolisation linéaire
```

```
'------------------------------------------------------------------------------
 Sub GereSymbPen

   Dialog
   title "choisissez un symbole linéaire"
  Control PenPicker
  Position 142,45
          Into Ligne
    ID 1

   Control OKButton
    ID 2

   Control CancelButton
    ID 3

        Set Style Pen Ligne

end sub
'============================================= FIN GERESYMBPEN
'============================================== GERESYMBZONE
'------------------------- COMMENTAIRES SUR LE MODULE --------------------------
' Procédure qui gère la symbolisation zonale
'------------------------------------------------------------------------------
 Sub GERESYMBZONE

   Dialog
   title "choisissez un symbole surfacique"
  Control BrushPicker
  Position 142,45
          Into Motif
    ID 1

   Control OKButton
    ID 2

   Control CancelButton
    ID 3

        Set Style Brush Motif

 end sub
'==================================== FIN GERESYMBZONE

'========================================== MenuSaisie
'------------------------- COMMENTAIRES SUR LE MODULE --------------------------
' Créée un panneau de boutons de saisie
'------------------------------------------------------------------------------
 Sub MenuSaisie

  create ButtonPad  "Styles" AS

   PushButton
     HelpMsg "Changer le style ponctuel"
     Calling GereSymbPonct
     Icon MI_ICON_SYMBOL_STYLE
          Separator
   PushButton
     HelpMsg "Changer le style linéaire"
     Calling GereSymbPen
     Icon MI_ICON_LINE_STYLE
          Separator
 PushButton
     HelpMsg "Changer le style zonal"
     Calling  GERESYMBZONE
     Icon MI_ICON_REGION_STYLE

 End Sub
'============================================== FIN MenuSaisie
```

X- Créer des dessins

En Map Info/Map Basic, l'accès au graphisme se réalise grace au type **object**. Les variables de type object peuvent être traitées comme les autres (affectées, passées en argument et bien sûr stockées dans des tables. La variable de type Object contient les informations géométriques et les informations de style (couleur...).Il y a une seule colonne « obj » par table.
On ne peut dessiner dans une fenêtre que si celle ci comporte une couche graphique active. Un objet (Object) est une forme qui appartient à l'un des types suivants: point, ligne, polyligne, arc, région, rectangle, rounded rectangle, ellipse et frame..
Map Basic peut créer de nouveaux objets, peut modifier des objets existants et ou faire des requêtes sur des objets.
Les objets peuvent être lus dans une table, stockés dans une variable du type Object ou encore écrits dans une table.

1-Les instructions de création

Création d'un Arc

Create Arc [**Into {Window** *Identificateur de fenêtre/* **Variable** *nom de variable* **}**]
(x1,y1)(x2,y2) Angle_début angle_fin [**Pen...**]

Création d'un point

Create Point [**Into {Window** *Identificateur de fenêtre/* **Variable** *nom de variable* **}**]
(x,y) [**Pen...**]

Création d'une Ligne

Create Line [**Into {Window** *Identificateur de fenêtre/* **Variable** *nom de variable* **}**]
(x1,y1)(x2,y2) [**Pen...**]

Création d'une zone

Create Region [**Into {Window** *Identificateur de fenêtre/* **Variable** *nom de variable* **}**]
Nombre_de_polygones
[nombre_de_point_du_premier_polygone (x1,y1)(x2,y2)[....(xn,yn)].]
[...]
[nombre_de_point_du_nième_pne (x1,y1)(x2,y2)[....(xn,yn)].]
[**Pen...**]
[**Brush...**]
[**Center (**Center_x, Center_y**)**]

Création d'une ou plusieurs régions par requête géométrique

Create Object As {Buffer / **Union/Intersect/Merge }**
from NomTable
[**Into {Table** *Identificateur de table/* **Variable** *nom de variable* **}**]
[**Width** *Buffer*[**Units** *Valeur d'unité*]]
[**Data** *colonne= expression* [,*colonne expression..*]]
[**Group by {** *colonne/* **RowId**]

Connaître les attributs d'un objet géomètrique
ObjectGeography ( ), ObjectNodeX ( ),ObjectNodeY ( ), etc...

ϟ ! Attention les objets de type Line (segment) et Pline (polyligne) NE SONT PAS MANIPULES PAR LES MEMES FONCTIONS. Ainsi ObjectGeography permettra de récupérer les coordonnées X et Y du début et de la fin d'un segment, mais pas d'une Polyligne.!!!, à contrario ObjectNodeX travaille sur les objets de type Region ou Pline.

EXERCICE 8: Savoir associer une géométrie à une table, savoir gérer les couches graphiques
Vous disposez dans le répertoire « données » de deux tables : « Emploi1 » et « Emploito ». ces tables contiennent des enregistrements concernant quelques grands aéroports internationaux. Année par année, elles indiquent le nombre de passagers et d'emplois sur chaque plate forme aéroportuaire.
Faire le programme Map Basic qui va dessiner les courbes , aéroport par aéroport, indiquant l'évolution emploi par passagers. (Milliers d'emplois en ordonnées et millions de passagers en abscisse). Gérer l'affichage du nom de chaque aéroport grâce aux étiquettes sous Map Info.

2- Corrigé de l'exercice de dessin de courbes d'emploi par passagers

```
'====================================================================
'            Module écrit en MAP-BASIC v 5.0.1      par Pascal BARBIER
'====================================================================
'
'Date de création                                    : jeudi 16 10 1997
'Date de dernière modification              : lundi 26 01 1998
'
'Nom du Fichier source: P200//c:\\_barbier\cours_mb\programs\exo_8.mb
'----------------------- COMMENTAIRES SUR LE MODULE -------------------------------
'------------------------------------------------------------------------------
' Déclaration des procédures
'------------------------------------------------------------------------------
 Declare Sub Main
 Declare Sub ChoisirTable (Tab As Integer, NomTabc As String, Message As String)
 Declare Sub Verificateur
 Declare Sub Dessine
 Declare Sub Bye

'============================================================ MAIN
'----------------------- COMMENTAIRES SUR LE MODULE -------------------------------
' Procédure principale
'------------------------------------------------------------------------------
 Sub Main
 Dim Table            As String

            include "..\DEF\MAPBASIC.DEF"
            include "..\DEF\ICONS.DEF"

    Create Menu "Dessin" As
             "Dessiner une table aéroport"
                HelpMsg          "trace les courbes Emploi/Passagers"
        Calling Dessine,
             "Supprimer Dessin"  Calling Bye
             Alter Menu Bar Add "Dessin"
 end sub
'============================================================ FIN MAIN

'============================================================ Bye
'----------------------- COMMENTAIRES SUR LE MODULE -------------------------
' Retire un menu déroulant de la barre de menus
'------------------------------------------------------------------------------
Sub Bye
     Alter Menu Bar remove "Dessin"
End Sub
'============================================================ FIN Bye

'============================================================ Dessinr
'------------------------- COMMENTAIRES SUR LE MODULE --------------------------
' procédure d'exploitation de dessin
'------------------------------------------------------------------------------
Sub Dessine

Dim Taille, i,k,j                As integer
Dim colpas,colemp,
   colaer,colan                  As Alias
Dim passager, emploi             As float
Dim Maxp,MinP,Maxe,MinE,
   X0, Y0,X1, Y1                 As Float
Dim nom                          As string
Dim Annee                        As Integer
Dim Aero                         As Object
Dim TabEmploi                    As string
Dim Tab                          s Integer
Dim NomTabDep,
   Message                       As String
Dim NbCol                        As Smallint
Dim Taillechar                   As float
Dim TabNom (200)                 As String
Dim NbAero                       As Integer
Dim Premier                      As Logical
Dim Symbole                      As Symbol
Dim Absc,Ordo                    As Integer
Message = "Choisissez la table a dessiner"
 Call ChoisirTable ( Tab, NomTabDep, Message)

'======== Choix des symbos
           symbole = MakeSymbol(44,RED,10) ‘ initialise le symbole carré rouge de taille 10
            Dialog        ' --- Choix du symbole
             Title "Symbole"
```

```
            Control StaticText
                    title "Symbole "
            Control SymbolPicker
              Position 140,142
              Value symbole
              ID 1
              into symbole
            Control OKButton
              ID 2
            Control CancelButton
              ID 3

   set style symbol Symbole
'============================== Gestion des points en coordonnées emplois/passagers
                                '- Création d'une couche géométrique
 Create Map for NomTabDep
 Map from NomTabDep
                                '
 Set Map Layer 1 Editable On 'Global Symbol Symbole

Taille = TableInfo (Tab, TAB_INFO_NROWS)
NbCol = TableInfo(NomTabDep,TAB_INFO_NCOLS)
NbAero = 0

  colpas = NomTabDep+".Passagers"
  colemp = NomTabDep+".Nb_emploi"
  colaer = NomTabDep+".nom"

  MinE= 999999                  '----Emploi Mini
  MinP= 999999                  '----Nb Passagers Mini
  MaxE = 0                              '----Emploi Maxi
  MaxP = 0                              '----Nb Passagers Maxi

  Fetch First from NomTabDep

    for i =1 to Taille
      passager = colpas
      emploi = colemp
      nom = colaer

      For k =1 to NbAero  '       --- NBAero contient le nb d'aéroports traités
        If TabNom (K)= nom then goto sortie
          end if
      next

      nbAero =NbAero+1              ' -- cas d'un nouvel aéroport
      TabNom (NbAero) = nom
      sortie:

      If Emploi> MaxE then MaxE= Emploi                 ' --- Recherche des min max
      elseif Emploi<MinE then MinE = Emploi
      end if
      If Passager> Maxp then Maxp= Passager
      elseif Passager<Minp then MinP = Passager
      end if
                        ' ---Création d'un objet géométrique ponctuel pour l'enregistrement
      update NomTabDep Set obj = CreatePoint(passager,emploi) where RowId =i

      Fetch next from NomTabDep
    next ' fin i

  Autolabel Window FrontWindow() Layer 1
'                       ------------ liaison entre aéroports
for j=1 to NbAero
 premier =false
 Fetch First from NomTabDep

 for i = 1 to Taille
      passager = colpas
      emploi = colemp
      nom = colaer

 if nom = TabNom(j)  then ' on tient un enregistrement concernant l'aéroport courant
   if premier = 0 then
     x0=passager
     y0 = emploi
     premier = 1
   else
    X1=passager
    y1 = emploi
    Create Line into Window FrontWindow() (X0,Y0)(X1,Y1)  pen (1,2,red)'-
    X0=X1
    Y0=Y1
  end if
```

```
 end if
     Fetch next from NomTabDep
 next ' fin i
next  ' fin j
'                          ---------- gestion des carroyages
TailleChar =(Maxe-Mine)/20
  Create Line into Window FrontWindow() (0,0)(0,Maxe+2*taillechar) '1
  Create Line into Window FrontWindow() (0,0)(Maxp+2*taillechar,0) '-
  Create Text into Window FrontWindow() " Milliers d'emplois"(Minp,Maxe+TailleChar)(Minp,Maxe+2*TailleChar)justify center
  Create Text into Window FrontWindow() " Millions de passagers par
an"(Maxp+TailleChar,Mine)(Maxp+TailleChar,Mine+TailleChar)justify left

  For Absc = 0 to Maxp+10 step 10
   Create Line into Window FrontWindow() (Absc,taillechar/2)(Absc,-taillechar/2)
   Create Text into Window FrontWindow() str$(Absc)(Absc,-taillechar)(Absc,-taillechar/2)justify center
  next
  For Absc = 0 to Maxe+10 step 10
   Create Line into Window FrontWindow() (-taillechar/2,Absc)(taillechar/2,Absc)
   Create Text into Window FrontWindow() str$(Absc)(-taillechar*2,Absc)(-taillechar,Absc+TailleChar/2)justify left
  next
'                          ------------------------Création de la ligne de pente
  Create Line into Window FrontWindow() (0,0)(Maxe,Maxe)  pen (1,2,green)'-
end sub

'============================================================= CHOISIRTABLE
'------------------------ COMMENTAIRES SUR LE MODULE ----------------------------
' Procédure servant à sélectionner une table  parmi les tables ouvertes. On note que les table EMPLOI1 et EMPLOITO qui ne
‘possèdent pas la même structure mais qui disposent chacune des trois champs utilisés se dessinent avec cette procédure
'-------------------------------------------------------------------------------
Sub ChoisirTable (Tab As Integer, NomTab As String, Message As String)

 Dim i As Integer
 Dim ChoixT, TabOpen(100) As String

'       ----------------------- >>>>>>>>> Vérification qu'il y ait une table ouverte
 If NumTables() = 0 Then note " Une table doit être ouverte !"
 End If

'       ------------------ >>>>>>>>> Sélection d'une table parmi les tables ouvertes
 For i =1 to NumTables()
  TabOpen (i) = TableInfo (i, TAB_INFO_NAME)
 next
 For i =1 to NumTables()
  ChoixT = ChoixT + TabOpen (i)+ ";"
 next
'       ---------------------------------------- >>>>>>>>> Construction du Dialogue
   Dialog
            Title "Choix de la TABLE"
     Control ListBox                    '-----1er controle
       Title  ChoixT
       Calling Verificateur
       ID 1

       Value 1
       Into Tab                        ' ------ Renvoie le numéro de la table
       Position 10, 92 Width 150 Height 80

     Control OKButton
       ID 2
       Disable

     Control CancelButton
       ID 3

     Control StaticText
       Position 5,10
       Width 160 Height 60
       Title  Message   "' Choisissez la table:"
      NomTab = tableInfo (Tab,TAB_INFO_NAME)' -- Transforme le nom en n° de table
end Sub
'========================================================= fin de DESSINE
'======================== VERIFICATEUR-- Voir exercices précédents
```

XI- Sélection SQL sur des tables

1-Bases de données - relations géométriques entre table

Il s'agit du cas ou la base de données n'est pas relationnelle. Chaque table s'appuie sur une géométrie. Pour trouver les relations entre les objets de la Table1 et les objets de la Table2, il faut connaître leur type géométrique et envisager la comparaison géométrique qui s'impose avec les opérateurs géométriques de Map Info.
Avantages: Pas besoins d'avoir une réflexion sur le modèle conceptuel de la base. Toute nouvelle donnée qui arrive avec une géométrie dans un système cartographique compatible peut être croisée avec une autre.
Inconvénients: A chaque questionnement il faut refaire le calcul ce qui peut être très long car la complexité des calculs augmente vite avec l'augmentation du nombre d'objets géométriques. De plus il faut soit géocoder les tables non géocodées, soit faire des jointures entre les tables sur la base d'un critère. Cela conduit à des tables avec trop de champs pour être manipulées correctement. Il faut donc réserver cette solution à des calculs peu fréquents sur un nombre d'objets réduit. (Par exemple Le MOS et la BD CARTO)

2-Bases de données - relations sémantiques entre table

Il s'agit du cas ou la base de données est relationnelle. Les relations importantes ont été identifiées lors de l'analyse conceptuelle, un choix d'implémentation a été fait. Des champs portent l'information relationnelle. Cela peut être un rang dans une table pour ménager des accès directs si l'accès se fait par programme ou bien une valeur quelconque si l'accès est réalisé par séquentiel indexé (clause select sur un champ index)
Avantages Optimisation et rapidité. Le calcul est effectué une fois pour toute. La rapidité d'accès est indépendante du nombre d'objet. Il n'y a pas redondance d'information. La base de données est plus compacte. On peut travailler directement sur les tables ACCESS, EXEL.. qui ne gèrent pas la géométrie.
Inconvénients Il faut connaître la structure de la base pour pouvoir l'exploiter pleinement. Map info ne sait pas exploiter ces relations sans passer par une jointure car dans ce cas on se retrouve avec une table unique.

3-Bases de données - relations sémantiques ET géométriques entre tables

C'est par exemple le cas dans la BDCarto entre les tronçons routiers et les sommets. On possède à la fois la relation sémantique et la géométrie de chacun des objets. Il faut donc choisir au cas par cas quelle est la meilleure solution.

4-Les principes de la sélection SQL

Une sélection est une opération qui consiste à extraire un certain nombre de colonnes sur un certain nombre de lignes appartenant à un certain nombre de tables à partir de certains critères sémantiques et/ou géométriques. Par ailleurs les sélections peuvent réaliser des opérations élémentaires sur les résultats comme compter le nombre d'enregistrements retenu, calculer la somme ou la moyenne d'une colonne sur l'ensemble des lignes. Souvent le résultat d'une sélection est placé dans une table.

La requête de base

Select *liste de noms de colonnes* **from** *nom de table*[, *nom de table*]
[**where** *groupe d'expression*] ' séparées par des AND ou des OR
[**into** *table destination*]
[**Group by** *liste de colonnes*]
[**Order by** *liste de colonnes*]

Les opérateurs ALL et ANY

Une requête SQL opère sur une table dite en entrée et produit une table en sortie. Il est donc possible de chaîner les requêtes SQL entre elles pour réaliser des requêtes complexes sur des structures relationnelles. Deux opérateurs **ALL** et **ANY** viennent simplifier l'articulation des requêtes SQL enchainées en précisant la clause **Where**
Any définit un sous domaine ou la clause **Where** va tester si une expression donnée est **VRAIE** pour au moins l'une des valeurs du sous domaine.
Au contraire all défini un sous domaine ou la clause **Where** va tester si une expression donnée est **VRAIE** pour toutes les valeurs du sous domaine.
Exemple: La requête suivante sélectionne tous les Clients dont le département d'origine est 46, 75 ou 87.
Select * From client Where departement = Any ("46", "75", "87")

Les requêtes comportant des nom de tables contenus dans des variables

Utilisez la syntaxe: **Run Command** « Select »+Identifiant+ « from » +NomTable + « where not obj »

ϟ ! Attention à respecter les blancs qui sépareront les constantes chaines de caractère et les variables lors de la reconstitution de la commande.
Exemple : si la variable Identifiant contient Com_Id au moment de l'appel on obtiendra selon les cas:
« Select »+Identifiant+ « from ».... donnera Select Com_Id from .... et s'exécutera
« Select »+Identifiant+ « from » .... donnera SelectCom_Idfrom.... et ne s'exécutera pas

EXERCICE 9: Faire des requêtes géométriques et sémantiques sur des tables portant des liens relationnels. Afficher les résultats

On dispose de la couche administrative de la BDCarto en particulier sous MapInfo (Table Commune) et d'une table EXEL nommée Accidents.XLS qui possède 4 champs

Champ Type : Type de l'accident 1 entre voitures 2 entre voiture et camion
3 entre voiture et vélo 4 entre voiture et piéton
Champ Nb_Blesses : nombre de blesses
Champ Nb_Morts : nombre de Morts
Champ Commune : Identificateur de la commune. Lien vers l'attribut ID_COM de la BDCarto.

Cette table possède autant d'enregistrements que d'accidents recensés. Écrire un programme qui:

-permet, par menu déroulant,(figure 1) d'afficher la carte des communes si elle n'est pas déjà affichée,

-permet de sélectionner le type d'accident, pour lequel on souhaite des informations, grâce à un menu case à cocher (figure 2

- permet de sélectionner une commune par clic souris sur la carte(puis sélection SQL géométrique) grâce à un bouton dédié (Figure 3)

-1 ère question) -qui affiche une table avec les enregistrement qui correspondent à la commune et au type d'accident retenu.(Sélection SQL sur attributs sémantiques)

-2 ème question)- Même programme, mais qui affiche dans une boite bloquante le nombre d'accident survenus le nombre de morts et de blessés pour la commune. (Sélection SQL avec clause Group by) (figure 4)

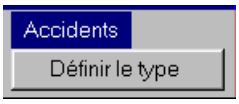


Figure 1

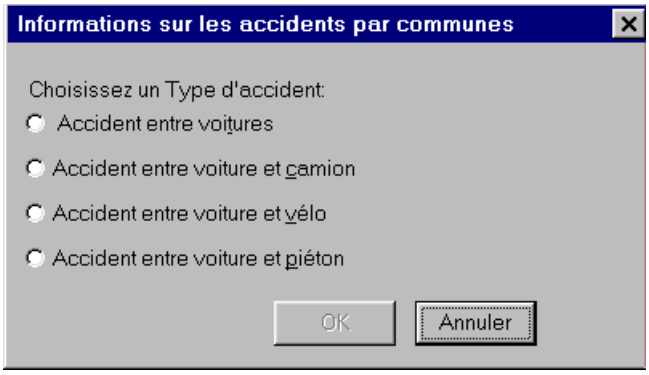


Figure 2

Figure 3

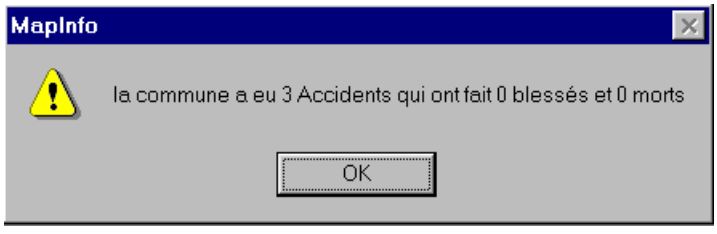


Figure 4

5-Corrigé de l'exercice sur les sélections SQL

```
'================================================================================
'      Programme écrit en MAP-BASIC V 5.0.1 par  :Pascal Barbier
'
'================================================================================
'
'Date de création                      : 28 novembre 1997
'Date de dernière modification   : 23 janvier 1998
'
'Nom du Fichier source: P200\\C:\_Barbier\cours_mb\programs\exo_9.mb
'------------------------- COMMENTAIRES SUR LE PROGRAMME --------------------------
'Exercice sur les requêtes
'--------------------------------------------------------------------------------

'--------------------------------------------------------------------------------
' Déclaration des procédures
'--------------------------------------------------------------------------------

 Declare Sub Main
 Declare Sub GereInterface
 Declare sub Getinfo
 Declare sub HandlerRadioGroup1
 Declare Sub Verificateur

global x As float
global Y As float
Global TypAcc As Integer

'============================================================================== MAIN
'------------------------- COMMENTAIRES SUR LE MODULE -----------------------------
' Procédure principale
'--------------------------------------------------------------------------------
 Sub Main

     include "..\DEF\MAPBASIC.DEF"
     include "..\DEF\ICONS.DEF"

       '--------------------------- Initialisation du nombre de points d'une région

     Create Menu "Accidents" As

      "Définir le type"
        HelpMsg
          "cliquer sur la commune"
           Calling GereInterface

      Alter Menu Bar Add "Accidents"

 end sub
'============================================================================ FIN MAIN

'=================================================================  HANDLERRADIOGROUP1

sub HandlerRadioGroup1

  Do Case readControlValue(1)
        case 1 TypAcc =1
        case 2 TypAcc =2
        case 3 TypAcc =3
        case 4 TypAcc =4
        case 5 Typacc =5
  end Case

  If ReadControlValue (1) then
         Alter Control 2 Enable
    else
         Alter Control 2 Disable
  End If

End Sub
'============================================================== FIN HANDLERRADIOGROUP1


'================================================================== GETINFO
 sub GetInfo

 Dim    localisation         As Object
 Dim   Identificateur        As Integer
 Dim   Nb_Rows                       As SmallInt
 Dim   ColId                         As Alias
 Dim   ColAcc,ColNbM,
               ColNbB, ColCom        As Alias
 Dim    NbAccidents,
               NbMorts,
```

```
               NbBlesses                    As smallint

' ---------------------------------Choisir la commune par clic

  Set Map coordsys Table Communes     ' ----permet de travailler dans le
                                      '     même système que la table Communes
  X = CommandInfo (CMD_INFO_X)
  Y = CommandInfo (CMD_INFO_Y)

  Create Point into Variable localisation (x,y)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'  select Id_Com from communes where Communes.obj contains localisation into TableTempo1
'  Nb_Rows = selectionInfo(SEL_INFO_NROWS)
'
'   If Nb_Rows = 1 then
'      colId = TableTempo1.ID_COM
'      Fetch Rec 1 from TableTempo1
'      Identificateur = ColId
'
'      ''''''''''''selection de la liste des accidents d'une commune (1ère question)
'      ''''    select * from Accidents where Accidents.commune = Identificateur  into TableResultat
                                            ''''
'      ''''''''''''affichage d'éléments d'information regroupés par commune (2ème question)
' If TypAcc<>5 then
'   select count(*),Sum(Nb_blesses),Sum(Nb_morts)
'              from Accidents
'              where Accidents.commune = Identificateur AND Accidents.Type= TypAcc
'              group by commune into TableResultat
'      else
'   select count(*),Sum(Nb_blesses),Sum(Nb_morts)
'              from Accidents
'              where Accidents.commune = Identificateur
'              group by commune into TableResultat
'      end If
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
       If TypAcc<>5 then
               select count(*),Sum(Nb_blesses),Sum(Nb_morts) from Accidents
               where Accidents.commune = ALL(select Id_Com from communes where Communes.obj
contains localisation ) AND Accidents.Type= TypAcc group by commune into TableResultat

               else
               select count(*),Sum(Nb_blesses),Sum(Nb_morts) from Accidents
                      where Accidents.commune = ALL(select Id_Com from communes where Communes.obj
                      contains localisation )group by commune into TableResultat

       end If

    Fetch Rec 1 from TableResultat
               ColAcc = TableResultat.COL1
               ColNbM = TableResultat.COL2
               ColNbB = TableResultat.COL3
               NbAccidents = ColAcc
               NbMorts=ColNbM
               NbBlesses=ColNbB

        Print CHR$(12)
       If TypAcc<>5 then
    print "La commune cliquée a eu "+NbAccidents+" Accidents du type"+TypAcc+","+CHR$(10)+"
               qui ont fait "
               +NbBlesses+" blessés et "+ NbMorts +" morts"
       else
    print "La commune cliquée a un total de "+NbAccidents+" Accidents de tout type,"+CHR$(10)+"
                      qui ont fait "
               +NbBlesses+" blessés et "+ NbMorts +" morts"

       end If
'''''''''' else note "Aucune commune trouvée! Recommencez!"
'''''''''' end if
 end sub

'========================================================================== FIN DE GETINFO

'===================================================================== GEREINTERFACE
'------------------------- COMMENTAIRES SUR LE MODULE -------------------------------
' Procédure qui gère l'interface utilisateur de l'outil de connaissance des accidents
'------------------------------------------------------------------------------------
 Sub GereInterface

 Dim    NumCom               As Integer
 Dim TabOpen (20)            As String
 Dim I                              As Integer
 Dim OuvAcc,OuvCom           As Logical

'---------------- ouvrir les tables

 OuvAcc = False
 OuvCom = False
```

```
For i =1 to NumTables()
 TabOpen (i) = TableInfo (i, TAB_INFO_NAME)
 If TabOpen (i) ="Accidents" then OuvAcc =True
 End if
 If TabOpen (i)= "Communes" then OuvCom =True
 End if
next

If Not OuvAcc then
  Open Table "..\donnees\Accidents" As Accidents
end if
If Not OuvAcc then
 Open Table "..\donnees\Communes" As Communes
 Map from communes

end if


Dialog
       Title " Informations sur les accidents par communes"
       Height 120
       Width 200

    Control RadioGroup
     Position 5,20
         ID 1
        calling HandlerRadioGroup1
        Title " Accident entre voi&tures;Accident entre voiture et &camion;Accident entre voiture
et &vélo;Accident entre voiture et &piéton; Tout Type d'accident"
        Into TypAcc
     Value TypAcc

     Control OKButton
       ID 2
       Disable

     Control CancelButton
       ID 3

     Control StaticText
       Position 5,10
       Width 160
       Title " Choisissez un Type d'accident:"


create ButtonPad  "Accidents" AS
 separator
 separator
 separator
 separator
 separator
 separator
 ToolButton
   HelpMsg "Informations par type d'accidents sur une commune"
   Calling GetInfo
   Icon 309 'MI_ICON_CROSSHAIR
   Cursor MI_CURSOR_FINGER_LEFT
   DrawMode  DM_CUSTOM_POINT

 separator
 separator
 separator
 separator

 position (1,1)
 Width 15



end sub
'=================================================================== FIN GEREINTERFACE
'==================================================== VERIFICATEUR
    Sub Verificateur

    If ReadControlValue (1) then
         Alter Control 2 Enable
    else
         Alter Control 2 Disable
    End If

        end sub
 '================================================ FIN VERIFICATEUR
```

XII-Lier MAP BASIC et MAP INFO

Intégrer une application MAP BASIC à MAP INFO

Nous venons de voir que, depuis l'environnement MapBasic, le l'option Run du menu Project bascule de l'environnement Map Basic vers le moteur MAP INFO. Ce n'est évidemment pas comme cela que les utilisateurs de vos logiciels vont pratiquer.
La première manière d'exécuter un Map Basic depuis Map Info est de choisir l'option Fichier / Exécuter....
Maintenant, si vous souhaitez qu'un programme Map Basic soit systématiquement chargé à l'appel de Mapinfo vous pouvez procéder de la manière suivante:

Par les propriétés de icône Map Info

Sous Windows 3.11: Modifiez les propriétés de MapInfo grâce à icône. Simple clic sur icône puis grâce au menu Propriété du gestionnaire de programme, remplacez la Ligne de commande qui doit ressembler à ceci:

C:\MapInfo\MapInfow.exe

par

C:\MapInfo\MapInfow.exe c:\programs\truc.mbx

Valider la modification par Ok. Désormais à chaque lancement de Map Info, le map Basic exécutable truc.mbx placé dans le répertoire programs sera exécuté.
Sous Windows 95 procéder de la même manière avec icône. Noter que dans ce cas le Dialogue de démarrage n'apparaîtra plus.

Avec le workspace spécial STARTUP

Startup est un nom de workspace spécial. Si ce fichier existe sur le système de l'utilisateur(dans le répertoire de MapInfo ou bien le répertoire de Windows), alors MapInfo le charge automatiquement. Si le Startup Workspace contient l'instruction : Run Application "truc.mbx" alors de la même manière que précédemment le MapBasic truc sera chargé au démarrage. Pour que MapInfo reconnaisse un fichier comme un workspace les trois première lignes suivantes sont indispensables.

!Workspace
!Version xxx (xxx est fonction de la version du compilateur Map Basic)
!Charset Neutral

Cette solution n'a pas d'effet sur l'affichage du dialogue de démarrage.

FIN

Annexes

MapBasic.Def (partiel)

Constantes définies par MAPBASIC.DEF

```
''' Logical constants '''''''''''''''''''''''''''''''''''''''''''''''''''''''
Define   TRUE          1
Define   FALSE         0

''' Angle conversion '''''''''''''''''''''''''''''''''''''''''''''''''''''''
Define DEG_2_RAD       .01745329252
Define RAD_2_DEG        57.29577951

''' Colors '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Define BLACK                    0
Define WHITE              16777215
Define RED               16711680
Define GREEN                 65280
Define BLUE                    255
Define CYAN                  65535
Define MAGENTA            16711935
Define YELLOW             16776960

''' TableInfo() defines '''''''''''''''''''''''''''''''''''''''''''''''''''''''
Define TAB_INFO_NAME        1
Define TAB_INFO_NUM         2
Define TAB_INFO_TYPE        3
Define TAB_INFO_NCOLS       4
Define TAB_INFO_MAPPABLE    5
Define TAB_INFO_READONLY    6
Define TAB_INFO_TEMP        7
Define TAB_INFO_NROWS       8
Define TAB_INFO_EDITED      9
Define TAB_INFO_FASTEDIT   10
Define TAB_INFO_UNDO       11

''' Table type defines, returned by TableInfo(<tab_ref>,
TAB_INFO_TYPE) '''''
Define TAB_TYPE_BASE        1
Define TAB_TYPE_RESULT      2
Define TAB_TYPE_VIEW        3
Define TAB_TYPE_IMAGE       4

''' ColumnInfo() defines ''''''''''''''''''''''''''''''''''''''''''''''''''''''
Define COL_INFO_NAME        1
Define COL_INFO_NUM         2
Define COL_INFO_TYPE        3
Define COL_INFO_WIDTH       4
Define COL_INFO_DECPLACES   5
Define COL_INFO_INDEXED     6
Define COL_INFO_EDITABLE    7

''' Column type defines, returned by ColumnInfo(<col_ref>,
COL_INFO_TYPE) '''
Define COL_TYPE_CHAR        1
Define COL_TYPE_DECIMAL     2
Define COL_TYPE_INTEGER     3
Define COL_TYPE_SMALLINT    4
Define COL_TYPE_DATE        5
Define COL_TYPE_LOGICAL     6
Define COL_TYPE_GRAPHIC     7
Define COL_TYPE_FLOAT       8

''' WindowInfo() defines '''''''''''''''''''''''''''''''''''''''''''''''''''''''
Define WIN_INFO_NAME        1
Define WIN_INFO_TYPE        3
Define WIN_INFO_WIDTH       4
Define WIN_INFO_HEIGHT      5
Define WIN_INFO_X           6
Define WIN_INFO_Y           7
Define WIN_INFO_TOPMOST     8
Define WIN_INFO_STATE       9
Define WIN_INFO_TABLE      10
Define WIN_INFO_OPEN       11

''' Window types, returned by WindowInfo(<win_id>,
WIN_INFO_TYPE) ''''''''''
Define WIN_MAPPER           1
Define WIN_BROWSER          2
Define WIN_LAYOUT           3
Define WIN_GRAPH            4
Define WIN_HELP          1001
Define WIN_MAPBASIC      1002
Define WIN_MESSAGE       1003
Define WIN_RULER         1007
Define WIN_INFO          1008
Define WIN_LEGEND        1009
Define WIN_STATISTICS    1010
Define WIN_MAPINFO       1011

''' Version 2 window types no longer used in version 3 '''''''''''''''''''''
Define WIN_TOOLPICKER    1004
Define WIN_PENPICKER     1005
Define WIN_SYMBOLPICKER  1006

''' Window states, returned by WindowInfo(<win_id>,
WIN_INFO_STATE) ''''''''
Define WIN_STATE_NORMAL     0
Define WIN_STATE_MINIMIZED  1
Define WIN_STATE_MAXIMIZED  2

''' ObjectInfo() defines ''''''''''''''''''''''''''''''''''''''''''''''''''''''
Define OBJ_INFO_TYPE        1
Define OBJ_INFO_PEN         2
Define OBJ_INFO_SYMBOL      2
Define OBJ_INFO_TEXTFONT    2
Define OBJ_INFO_BRUSH       3
Define OBJ_INFO_NPNTS      20
Define OBJ_INFO_TEXTSTRING  3
Define OBJ_INFO_SMOOTH      4
Define OBJ_INFO_FRAMEWIN    4
Define OBJ_INFO_NPOLYGONS  21
Define OBJ_INFO_TEXTSPACING 4
Define OBJ_INFO_TEXTJUSTIFY 5
Define OBJ_INFO_FRAMETITLE  6
Define OBJ_INFO_TEXTARROW   6

''' Object types, returned by ObjectInfo(<obj>, OBJ_INFO_TYPE)
''''''''''''''
Define OBJ_ARC              1
Define OBJ_ELLIPSE          2
Define OBJ_LINE             3
Define OBJ_PLINE            4
Define OBJ_POINT            5
Define OBJ_FRAME            6
Define OBJ_REGION           7
Define OBJ_RECT             8
Define OBJ_ROUNDRECT        9
Define OBJ_TEXT            10

''' ObjectGeography() defines '''''''''''''''''''''''''''''''''''''''''''''''''
Define OBJ_GEO_MINX         1
Define OBJ_GEO_LINEBEGX     1
Define OBJ_GEO_POINTX       1
Define OBJ_GEO_MINY         2
Define OBJ_GEO_LINEBEGY     2
Define OBJ_GEO_POINTY       2
Define OBJ_GEO_MAXX         3
Define OBJ_GEO_LINEENDX     3
Define OBJ_GEO_MAXY         4
Define OBJ_GEO_LINEENDY     4
Define OBJ_GEO_ARCBEGANGLE  5
Define OBJ_GEO_TEXTLINEX    5
Define OBJ_GEO_ROUNDRADIUS  5
Define OBJ_GEO_ARCENDANGLE  6
Define OBJ_GEO_TEXTLINEY    6
Define OBJ_GEO_TEXTANGLE    7

''' StyleAttr() defines '''''''''''''''''''''''''''''''''''''''''''''''''''''''
Define PEN_WIDTH            1
Define PEN_PATTERN          2
Define PEN_COLOR            4
Define BRUSH_PATTERN        1
Define BRUSH_FORECOLOR      2
Define BRUSH_BACKCOLOR      3
Define FONT_NAME            1
Define FONT_STYLE           2
Define FONT_POINTSIZE       3
Define FONT_FORECOLOR       4
Define FONT_BACKCOLOR       5
Define SYMBOL_CODE          1
Define SYMBOL_COLOR         2
Define SYMBOL_POINTSIZE     3

''' MapperInfo() defines ''''''''''''''''''''''''''''''''''''''''''''''''''''''
Define MAPPER_INFO_ZOOM     1
```

```
Define MAPPER_INFO_SCALE      2
Define MAPPER_INFO_CENTERX    3
Define MAPPER_INFO_CENTERY    4
Define MAPPER_INFO_MINX       5
Define MAPPER_INFO_MINY       6
Define MAPPER_INFO_MAXX       7
Define MAPPER_INFO_MAXY       8
Define MAPPER_INFO_LAYERS     9
Define MAPPER_INFO_EDIT_LAYER 10
Define MAPPER_INFO_XYUNITS    11
Define MAPPER_INFO_DISTUNITS  12
Define MAPPER_INFO_AREAUNITS  13
Define MAPPER_INFO_SCROLLBARS 14

''' LayerInfo() defines ''''''''''''''''''''''''''''''''''''''''''''''
Define LAYER_INFO_NAME          1
Define LAYER_INFO_EDITABLE      2
Define LAYER_INFO_SELECTABLE    3
Define LAYER_INFO_ZOOM_LAYERED   4
Define LAYER_INFO_ZOOM_MIN      5
Define LAYER_INFO_ZOOM_MAX      6
Define LAYER_INFO_COSMETIC      7
Define LAYER_INFO_PATH          8
Define LAYER_INFO_DISPLAY       9
Define LAYER_INFO_OVR_LINE      10
Define LAYER_INFO_OVR_PEN       11
Define LAYER_INFO_OVR_BRUSH     12
Define LAYER_INFO_OVR_SYMBOL    13
Define LAYER_INFO_OVR_FONT      14
Define LAYER_INFO_LBL_EXPR      15
Define LAYER_INFO_LBL_LT        16
Define LAYER_INFO_LBL_CURFONT   17
Define LAYER_INFO_LBL_FONT      18
Define LAYER_INFO_LBL_PARALLEL  19
Define LAYER_INFO_LBL_POS       20
Define LAYER_INFO_ARROWS        21
Define LAYER_INFO_NODES         22
Define LAYER_INFO_CENTROIDS     23
Define LAYER_INFO_TYPE          24

''' Display Modes, returned by LayerInfo() for LAYER_INFO_DISPLAY
''''''''''
Define LAYER_INFO_DISPLAY_OFF        0
Define LAYER_INFO_DISPLAY_GRAPHIC    1
Define LAYER_INFO_DISPLAY_GLOBAL     2
Define LAYER_INFO_DISPLAY_VALUE      3

''' Label Linetypes, returned by LayerInfo() for LAYER_INFO_LBL_LT
'''''''''
Define LAYER_INFO_LBL_LT_NONE    0
Define LAYER_INFO_LBL_LT_SIMPLE  1
Define LAYER_INFO_LBL_LT_ARROW   2

''' Label Positions, returned by LayerInfo() for
LAYER_INFO_LBL_POS ''''''''
Define LAYER_INFO_LBL_POS_CC   0
Define LAYER_INFO_LBL_POS_TL   1
Define LAYER_INFO_LBL_POS_TC   2
Define LAYER_INFO_LBL_POS_TR   3
Define LAYER_INFO_LBL_POS_CL   4
Define LAYER_INFO_LBL_POS_CR   5
Define LAYER_INFO_LBL_POS_BL   6
Define LAYER_INFO_LBL_POS_BC   7
Define LAYER_INFO_LBL_POS_BR   8

''' Layer Types, returned by LayerInfo() for LAYER_INFO_TYPE
'''''''''''''''
Define LAYER_INFO_TYPE_NORMAL    0
Define LAYER_INFO_TYPE_COSMETIC  1
Define LAYER_INFO_TYPE_IMAGE     2
Define LAYER_INFO_TYPE_THEMATIC  3

''' CommandInfo() defines ''''''''''''''''''''''''''''''''''''''''''''''''
Define CMD_INFO_X          1
Define CMD_INFO_Y          2
Define CMD_INFO_SHIFT      3
Define CMD_INFO_CTRL       4
Define CMD_INFO_X2         5
Define CMD_INFO_Y2         6
Define CMD_INFO_TOOLBTN    7
Define CMD_INFO_MENUITEM   8
Define CMD_INFO_WIN        1
Define CMD_INFO_SELTYPE    1
Define CMD_INFO_ROWID      2
Define CMD_INFO_STATUS     1
Define CMD_INFO_MSG        1000
Define CMD_INFO_DLG_OK     1
Define CMD_INFO_DLG_DBL    1
Define CMD_INFO_FIND_RC    3
Define CMD_INFO_FIND_ROWID  4
Define CMD_INFO_XCMD       1

''' SelectionInfo() defines ''''''''''''''''''''''''''''''''''''''''''''
Define SEL_INFO_TABLENAME   1
Define SEL_INFO_SELNAME     2
Define SEL_INFO_NROWS       3

''' Return Values from StringCompare(<str_1>, <str_2>) ''''''''''''''''''''
Define STR_LT            -1
Define STR_GT             1
Define STR_EQ             0

''' Parameters used for IntersectNodes(obj1, obj2, mode) ''''''''''''''''''
Define INCL_CROSSINGS      1
Define INCL_COMMON         6
Define INCL_ALL          7

''' Macros ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Define CLS Print Chr$(12)
```

Icons.Def

```
'*---------------------------------------------------------------------*
' Defines for cursors built into MapBasic.
' Available to be used with the ButtonPads.
'*---------------------------------------------------------------------*
define MI_CURSOR_ARROW          0
define MI_CURSOR_IBEAM          1
define MI_CURSOR_FINGER_LEFT    128
define MI_CURSOR_ZOOM_IN        129
define MI_CURSOR_ZOOM_OUT       130
define MI_CURSOR_DRAG_OBJ       131
define MI_CURSOR_GRABBER        132
define MI_CURSOR_CHANGE_WIDTH   133
define MI_CURSOR_FINGER_UP      134
define MI_CURSOR_IBEAM_CROSS    135
define MI_CURSOR_CROSSHAIR      138
'*-------------------------------------------------------*
' Defines for different DrawModes for the custom tool.
'*-------------------------------------------------------*
define DM_CUSTOM_CIRCLE     30
define DM_CUSTOM_ELLIPSE    31
define DM_CUSTOM_RECT       32
define DM_CUSTOM_LINE       33
define DM_CUSTOM_POINT      34
'*---------------------------------------------------------------------*
' Defined constants for icons built into MapInfo.
' Available to be used on the ButtonPads.
'*---------------------------------------------------------------------*
define MI_ICON_ARROW          0
define MI_ICON_SEARCH_RECT    1
define MI_ICON_SEARCH_RADIUS  2
define MI_ICON_SEARCH_BDY     3
define MI_ICON_ZOOM_IN        4
define MI_ICON_ZOOM_OUT       5
define MI_ICON_ZOOM_QUESTION  6
define MI_ICON_GRABBER        7
define MI_ICON_INFO           8
define MI_ICON_LABEL          9
define MI_ICON_LAYERS         10
define MI_ICON_RULER          11
define MI_ICON_LEGEND         12
define MI_ICON_STATISTICS     13
define MI_ICON_DISTRICT_MANY  14
define MI_ICON_DISTRICT_SAME  15
define MI_ICON_SYMBOL         16
define MI_ICON_LINE           17
define MI_ICON_POLYLINE       18
define MI_ICON_ARC            19
define MI_ICON_POLYGON        20
define MI_ICON_ELLIPSE        21
define MI_ICON_RECT           22
define MI_ICON_ROUND_RECT     23
define MI_ICON_TEXT           24
define MI_ICON_WINDOW_FRAME   25
define MI_ICON_RESHAPE        26
define MI_ICON_ADD_NODE       27
define MI_ICON_SYMBOL_STYLE   28
define MI_ICON_LINE_STYLE     29
define MI_ICON_REGION_STYLE   30
define MI_ICON_TEXT_STYLE     31
define MI_ICON_RUN            32
define MI_ICON_WRENCH         33
define MI_ICON_CROSSHAIR      34
```