

BreakPoint pour TI89 / T92 Plus / V200

V1.00

Copyright © 2008, François Vuillet

TABLE DES MATIÈRES

PRESENTATION.....	2
Avantages de BreakPoint.....	2
INSTALLATION.....	2
1-Compatibilité.....	2
2-Taille.....	2
3-Chargement du fichier.....	2
UTILISATION.....	2
1-Exemple.....	2
2-Insertion de la commande brk\pt dans un programme.....	3
a)Nommer le point d'insertion de la commande brk\pt.....	3
b)Indiquer des variables globales à inspecter.....	4
c)Indiquer des expressions ne contenant pas de variables locales.....	4
d)Indiquer des variables locales ou des expressions contenant des variables locales.....	5
e)Effacer la liste des variables précédemment inspectées.....	5
f)Interruption du programme à déboguer juste après l'exécution de BreakPoint.....	5
3-Déroulement de BreakPoint.....	5
a)Sélection de l'expression à visualiser.....	5
b)Ajout d'une expression à visualiser.....	6
c)Utilisation de la boîte de dialogue d'édition dans le cas d'une variable globale.....	6
d)Utilisation de la boîte de dialogue visualisation d'une expression.....	7
VARIABLES GLOBALES ET VARIABLES LOCALES.....	8
LIMITATIONS.....	8
BUGS CONNUS.....	8
VARIABLES UTILISEES PAR LE PROGRAMME.....	8
MENU PERSONNALISE.....	8
REMERCIEMENTS.....	8
CONTACTER L'AUTEUR.....	9

BreakPoint pour TI89 / T92 Plus / V200

PRESENTATION

BreakPoint est un utilitaire de mise au point pour les programmes écrits en BASIC. Il est écrit en BASIC et a été validé sur Ti-89 Titanium.

BreakPoint imite le fonctionnement du point d'arrêt dont on dispose dans un débogueur. Il permet de suspendre l'exécution du programme à déboguer, d'inspecter ses variables et au besoin de les modifier; pour cela il suffit d'insérer la commande **brk\pt** dans le programme à déboguer, aux endroits à analyser.

Généralement pour déboguer un programme on fait appel aux instructions *DISP* et *PAUSE* pour afficher le contenu de la variable à vérifier ; cette méthode a l'inconvénient d'entraîner le plantage du programme si la variable est effacée par exemple.

Avantages de BreakPoint

Si le contenu de la variable ne peut être mis sous forme de texte, par exemple variable de type *DATA*, *PRGM* ou *GDB*, **BreakPoint** fournit un minimum d'informations sur la variable et l'exécution du programme peut continuer.

Le point d'insertion de la commande **brk\pt** est nommé ce qui permet de savoir dans quelle partie du programme a lieu l'arrêt.

Le nom de la variable visualisée est indiqué en même temps que son contenu.

Lors de l'exécution de **BreakPoint** il est possible de visualiser d'autres variables que celles qu'on a prévues lors de l'insertion de la commande **brk\pt**, sans avoir besoin d'arrêter et de modifier le programme à déboguer.

INSTALLATION

1- Compatibilité

BreakPoint fonctionne uniquement sous les OS en version anglaise.

TI-89 avec AMS 2.01 minimum

TI-89 Titanium

TI-92 Plus avec AMS 2.03 minimum

Voyage™ 200

2- Taille

15Ko

3- Chargement du fichier

Charger le fichier *brkpt* sur votre calculatrice (.89p pour TI-89, .9xp pour TI-92 Plus ou .v2p pour V200).

Le programme *pt*, issu du fichier *brkpt*, doit être mis exclusivement dans le dossier *BRK*.

UTILISATION

1- Exemple

Regardons l'utilisation de **BreakPoint** au travers d'un exemple simple.

Le programme suivant affiche à l'écran un avion et le déplace de gauche à droite. A chaque pas de déplacement de l'avion l'exécution de **BreakPoint** permet d'inspecter les différentes variables du programme. Au bout de quelques pas, au moment de l'exécution de **BreakPoint**, on attribuera la valeur 150 à la variable *horz* pour abrégier l'exécution du programme.

BreakPoint pour TI89 / T92 Plus / V200

```
plane()
Prgm
setFold(example)
DelVar horz,vert,lasth,lastv,lft,right
brk\pt("", "horz,vert,lasth,lastv,lft,right") <----- Choix des variables à inspecter
ClrIO
10→lft
150→right
10→lasth
30→lastv
30+iPart(10*sin(.045*horz))→vert

For horz,lft,right,2
  brk\pt("loop","") <----- Inspection des variables
  Output lastv,lasth," "
  Output vert,horz,char(26)
  horz→lasth
  vert→lastv
EndFor

DispHome
EndPrgm
```

Ce programme d'exemple est fourni dans le fichier *plane*. Transférer le fichier correspondant à votre calculatrice et lancer *example\plane()*.

Lors de l'exécution du programme d'exemple, à chaque passage par la commande **brk\pt("loop", "")**, le programme **BreakPoint** est lancé :

La liste des variables visualisables s'affiche. Sélectionner la ligne 7



La variable sélectionnée est visualisée :



exemple\plane()
Taper [ESC] pour revenir à la liste des variables.

Sortir de **BreakPoint** en tapant [ESC] pour permettre au programme *plane* d'exécuter plusieurs cycles.

Afin d'abréger le déroulement de *plane*, modifier la variable *horz*:

Sélectionner la variable *horz* puis saisir 150 puis [ENTER] sous la ligne *Value*.



exemple\plane()
puis taper une deuxième fois [ENTER] et ensuite [ESC].

2- Insertion de la commande brk\pt dans un programme

Placer la commande **brk\pt(point,expressions)** dans le programme à déboguer aux endroits à inspecter.

Les arguments **point** et **expressions** sont expliqués ci-après.

a) Nommer le point d'insertion de la commande brk\pt

Le 1er argument de la commande **brk\pt** est une information permettant de se repérer lors de l'exécution du programme à déboguer.

Cet argument se trouvera affiché dans le titre des boîtes de dialogue et en première ligne des popups.

BreakPoint pour TI89 / T92 Plus / V200

Exemple :

```
brk\pt("loop", "horz,vert,lasth,lastv,lft,right")
```



Attention : Une chaîne vide à la place du nom du point d'insertion de la commande **brk\pt** commande l'effacement de la liste permanente des variables globales précédemment inspectées.

b) Indiquer des variables globales à inspecter

La liste des variables globales à inspecter est passée en 2^e argument de la commande **brk\pt**. Les variables sont séparées par des virgules, l'ensemble est mis entre guillemets. Si le chemin d'accès de la variable n'est pas indiqué la variable est considérée comme appartenant au dossier courant au moment de l'appel de **BreakPoint**.

Exemple :

```
brk\pt("loop ", " horz,vert,math\p")
```

Si au moment de l'appel de **BreakPoint** le dossier courant est *EXAMPLE* les variables *horz* et *vert* seront explicitées en *example\horz* et *example\vert*.

A chaque exécution de **BreakPoint** les noms des variables globales à inspecter sont ajoutés à une liste permanente et conservés jusqu'à l'exécution suivante. Les variables inspectées antérieurement sont ainsi proposées lors de l'exécution des commandes **brk\pt** qui suivent.

Par exemple si à chaque exécution de **BreakPoint** on souhaite inspecter uniquement les variables globales *horz*, *vert*, *lasth*, *lastv*, *lft* et *right*, la liste des variables ne sera passée en argument que de la première commande **brk\pt** :

```
...
brk\pt("Point #1", "horz,vert,lasth,lastv,lft,right")
...
brk\pt("Point #2", "") <----- Les variables horz, vert,
                           lasth,lastv, lft et right
...
brk\pt("Point #3", "") <----- seront à nouveau proposées
                           lors des exécutions
...
brk\pt("Point #4", "") <----- suivantes de BreakPoint.
...
```

Les variables passées en argument de la commande **brk\pt** en cours d'exécution seront repérées par une coche dans le popup de sélection de variables.

c) Indiquer des expressions ne contenant pas de variables locales

Les expressions à inspecter, à condition qu'elles ne contiennent pas de variables locales, sont passées en 2^e argument de la commande **brk\pt** de cette manière:

Les expressions sont placées dans une matrice dont chaque ligne contient l'expression mise entre guillemets dans la 1^{re} colonne, suivie d'un 0 dans la 2^e colonne.

Remarque : une expression peut se limiter à une variable globale.

Exemple :

```
brk\pt("wave1", [[ "3+sin(θ)", 0 ] [ "θ", 0 ] ])
```

BreakPoint pour TI89 / T92 Plus / V200

Si l'expression est une fonction, 2 solutions :

- `brk\pt("function",["f",∅])`
Lors de son exécution **BreakPoint** affichera la fonction en proposant arbitrairement les arguments a, b, c...
- `brk\pt("function",["f(x,y,z)",∅])`
Lors de son exécution **BreakPoint** affichera la fonction en utilisant les arguments x, y et z.

d) Indiquer des variables locales ou des expressions contenant des variables locales

BreakPoint ne peut pas accéder aux variables locales du programme à déboguer aussi il est nécessaire de passer la valeur de la variable ou de l'expression sous forme de chaîne de caractères en 2^e argument de la commande **brk\pt**.

Les variables et expressions sont placées dans une matrice dont chaque ligne contient le nom de la variable ou l'expression mise entre guillemets dans la 1^{re} colonne, suivi de sa valeur sous forme de chaîne de caractères dans la 2^e colonne.

Exemple:

Local w,t

...

`brk\pt("wave2",["3+sin(w*t)",string(3+sin(w*t))]["t",string(t)])`

Attention : Les variables et les expressions doivent pouvoir être converties en chaînes de caractères, donc ce n'est pas valable avec une variable de type DATA par exemple

e) Effacer la liste des variables précédemment inspectées

A chaque exécution de **BreakPoint** les noms des variables globales à inspecter sont ajoutés à une liste permanente et conservés jusqu'à l'exécution suivante.

Pour réinitialiser cette liste mettre une chaîne vide en 1^{er} argument de la commande **brk\pt** et indiquer éventuellement la nouvelle liste des variables globales en 2^e argument.

Exemples :

`brk\pt("", "")` : Efface la liste permanente des noms de variables globales.

`brk\pt("", "i,j,k")` : Réinitialise la liste permanente des noms de variables globales avec uniquement i, j et k pour les inspecter à la prochaine exécution de **BreakPoint**.

Remarque : Cette commande n'engendre pas l'inspection des variables lors de l'exécution.

f) Interruption du programme à déboguer juste après l'exécution de BreakPoint

`brk\pt(..., ...)` : PAUSE

L'instruction PAUSE permet d'éditer le programme à déboguer à la ligne en cours après une interruption par appui sur la touche ON.

3- Déroulement de BreakPoint

a) Sélection de l'expression à visualiser

Les expressions (une expression peut être une variable) indiquées en argument de la commande **brk\pt** ainsi que toutes les variables globales des commandes **brk\pt** précédentes sont affichées. Les expressions indiquées à la commande **brk\pt** en cours sont cochées.

Le nom du point d'insertion de la commande **brk\pt** est indiqué en première ligne de la liste.

BreakPoint pour TI89 / T92 Plus / V200

Exemple :

Commandes **brk\pt** dans le programme à déboguer:

```
brk\pt("", "lasth, lastv, lft")
...
brk\pt("loop", "horz, vert")
```

Liste des expressions à visualiser:

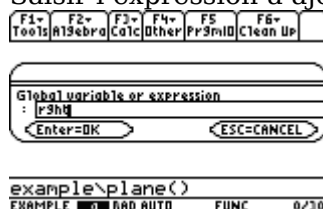


b) Ajout d'une expression à visualiser

D'autres expressions peuvent être ajoutées dans la liste de sélection en allant sur la dernière ligne "New expression":

Exemple:

Sélectionner la dernière ligne: Saisir l'expression à ajouter:



L'expression est maintenant dans la liste:



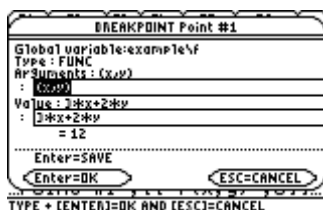
Lorsqu'une seule expression est indiquée en argument de la commande **brk\pt**, le sélecteur n'est pas affiché et **BreakPoint** passe directement à la visualisation de l'expression.

L'appui sur **[ESC]** lorsque la liste des expressions est affichée permet de sortir de **BreakPoint** et de reprendre l'exécution du programme en cours de débogage.

c) Utilisation de la boîte de dialogue d'édition dans le cas d'une variable globale

Le nom du point d'insertion de la commande **brk\pt** est indiqué dans le titre de la fenêtre.

Différentes informations sur la variable sont affichées :



- En première ligne est indiqué s'il s'agit d'une variable globale ou d'une variable système ainsi que le nom de la variable.
- En seconde ligne est indiqué le type de la variable.
- Si la variable est une fonction ses arguments sont affichés. Ils peuvent être modifiés dans la ligne en dessous; ils doivent être mis entre parenthèses. Dans l'exemple précédent, la variable est la fonction $f(x,y)$.
Si les arguments de la fonction n'ont pas été indiqués lors de l'insertion de la commande **brk\pt** dans le programme à déboguer, **BreakPoint** les nomme arbitrairement a, b, c, d... En éliminant les arguments et les parenthèses la variable cesse d'être une fonction. Dans le cas d'une variable vide, la ligne d'édition des arguments donne la possibilité de définir une nouvelle fonction.
- Si elle est visualisable, la valeur de la variable est ensuite affichée (évaluation de premier niveau). Elle peut être modifiée dans la ligne en dessous.

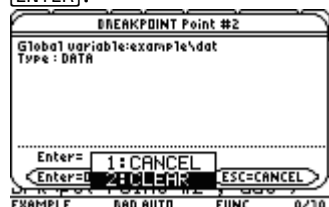
BreakPoint pour TI89 / T92 Plus / V200

Sur la ligne suivante est affichée l'évaluation totale si elle diffère de l'évaluation de premier niveau. Si l'évaluation totale n'est pas réalisable un message d'erreur est affiché à la place.

Les chaînes de caractères sont saisies et affichées exactement de la même manière qu'elles apparaissent dans la ligne de saisie de l'écran HOME.

Pour effacer la variable laisser vide la ligne de saisie de la valeur.

Si la valeur de la variable n'est pas affichée l'effacement de la variable est cependant possible en sélectionnant CLEAR dans le sélecteur en bas de la fenêtre puis en tapant [ENTER]:



La tentative de modification ou d'effacement d'une variable verrouillée ou archivée se traduit par un message d'erreur.

d) Utilisation de la boîte de dialogue visualisation d'une expression

Exemples :

1- Expression globale

Dans le programme à déboguer:

3→b

...
brk\pt("Point #3",[["2b",Ø]])

Déroulement du point d'arrêt :



2- Expression locale

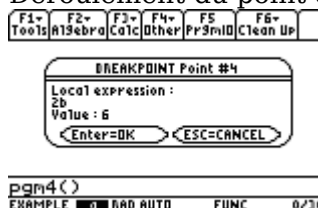
Dans le programme à déboguer:

Local b

...
3→b

...
brk\pt("Point #4",[["2b",string(2b)]])

Déroulement du point d'arrêt :



- En première ligne est indiqué s'il s'agit d'une expression locale ou globale (une expression locale peut être une variable locale).
- En seconde ligne est indiquée l'expression sélectionnée.
- S'il s'agit d'une expression globale:
Sur la troisième ligne est affichée l'évaluation de l'expression (évaluation de premier niveau). Si l'évaluation n'est pas réalisable un message d'erreur est affiché à la place.
Sur la ligne suivante est affichée l'évaluation totale si elle diffère de l'évaluation de premier niveau. Si l'évaluation totale n'est pas réalisable un message d'erreur est affiché à la place.
Les chaînes de caractères sont saisies et affichées exactement de la même manière qu'elles apparaissent dans la ligne de saisie de l'écran HOME.
- S'il s'agit d'une expression locale:
Sur la troisième ligne est affichée l'évaluation passée en 2^e argument de la commande **brk\pt**.

BreakPoint pour TI89 / T92 Plus / V200

VARIABLES GLOBALES ET VARIABLES LOCALES

Le système d'exploitation TI définit deux types de variables: local et global. Les variables locales sont déclarées dans les programmes ou les fonctions avec l'instruction *Local*, toutes les autres variables sont globales.

Une variable locale est une variable temporaire qui n'existe que pendant l'exécution d'un programme ou d'une fonction. A l'arrêt du programme les variables locales sont automatiquement effacées.

Si un programme possède des variables locales, un autre programme ne peut pas accéder à celles-ci.

Lors du débogage, il ne faut pas utiliser de variables locales afin que **BreakPoint** puisse accéder à la valeur des variables.

La calculatrice ne peut pas utiliser de variables locales pour réaliser des calculs symboliques.

LIMITATIONS

Breakpoint n'est pas utilisable dans les fonctions : transformer temporairement les fonctions en programmes.

Breakpoint ne peut pas lire ou modifier le contenu des variables locales ; il faut passer le contenu de la variable en argument de la commande **brk\pt** ou déclarer les variables comme globales.

Breakpoint ne fonctionne qu'avec l'OS en anglais.

BUGS CONNUS

Une variable contenue dans l'expression inspectée, ou la variable inspectée elle-même, peut devenir inaccessible en lecture si elle contient une variable non algébrique (type DATA par exemple). Il faut alors sortir du programme et générer une erreur : par exemple taper ☐ puis ENTER dans l'écran HOME. La variable redevient alors accessible.

VARIABLES UTILISEES PAR LE PROGRAMME

BreakPoint utilise la variable BRK\ælstvars pour stocker la liste des variables inspectées lors de ses exécutions successives. **BreakPoint** utilise temporairement le dossier BRKBLANK afin d'évaluer les expressions, ce dossier est automatiquement créé puis effacé par **BreakPoint**. Les dossiers BRKBLANK et ïïïðñòó ne doivent pas exister au lancement de **BreakPoint**.

MENU PERSONNALISE

Le menu personnalisé **BreakPoint** facilite l'insertion de la commande **brk\pt** dans les programmes à déboguer. Charger le fichier **menu** correspondant à votre calculatrice et exécuter **brk\menu()**.

REMERCIEMENTS

Merci à :

- L'équipe TiEmu pour leur émulateur.
- David Fernando Suescun Ramirez pour son programme Daisuke-Edit.

Ce manuel a été écrit avec la suite OpenOffice.org en utilisant les polices de caractères Bitstream DejaVu Serif, TI-89 symbols et Ti89pc.

BreakPoint pour TI89 / T92 Plus / V200

CONTACTER L'AUTEUR

Envoyez-moi un e-mail à francois.vuillet@laposte.net

Si vous trouvez un bug, si vous avez des suggestions, N'HESITEZ PAS ! J'essaierai de répondre à tous vos mails.

J'espère que ce programme vous aidera.

François VUILLET