

# UNE MIRE POUR CALCULATRICE

Auteur : Jean-Baptiste Civet

TI-83 Premium CE

**Mots-clés :** représentation graphique, minimum, maximum, programmation graphique.

**Fichiers associés :** JeuDeTir\_fiche1\_mire\_eleve.pdf, DINIT.8XP, DMIRE.8XP, DMIRE2.8XP

## 1. Objectifs

- Apprendre à utiliser les primitives graphiques spécifiques à la TI-83 Premium CE dans l'environnement de programmation de la calculatrice.
- Utiliser les notions sur les fonctions pour construire des dessins dans l'écran graphique.
- Il s'agit de la première activité d'un ensemble de trois activités aboutissant à la programmation d'un jeu fonctionnel.

## 2. Énoncé



Pendant quelques décennies, certaines chaînes de télévision diffusaient, à la fin de leurs programmes chaque soir, une mire, c'est-à-dire une image fixe composée de différentes formes géométriques (quadrillage et cercle essentiellement) et de différentes couleurs.

Cette mire permettait notamment le réglage des appareils à réception analogique.

Ce que l'on propose dans cette première activité est de produire une mire adaptée à la TI-83 Premium CE.

[http://commons.wikimedia.org/wiki/File:Philips\\_PM5544.svg](http://commons.wikimedia.org/wiki/File:Philips_PM5544.svg)

## 3. Commentaires

L'enjeu de cette activité est fondamental avant d'aborder la programmation du jeu dans les activités 2 et 3 suivantes.

Il y a, bien sûr, un enjeu technique pour que l'élève s'approprie l'environnement de programmation graphique de la TI-83 CE.

Il sera important de s'assurer notamment de la bonne maîtrise des éléments suivants :

- Modalité de définition d'une fonction et de l'usage des guillemets.
- Calcul de valeurs d'une fonction dans l'environnement de programmation.
- Compréhension des paramètres numériques de la définition de la couleur ([10-24]).
- Compréhension de la notion de pixel et des dimensions de l'écran de la TI-83 CE (265x165).

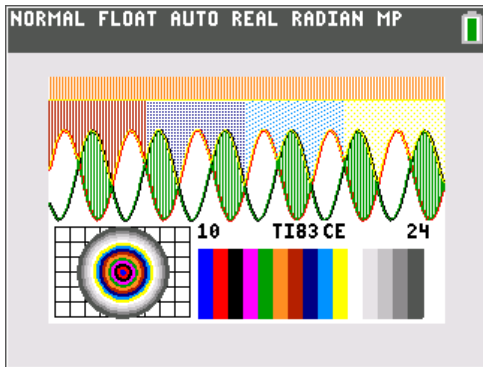
Il y a, également, un enjeu mathématique, autour du dessin de la mire circulaire et de l'usage de la définition « fonctionnelle » d'un cercle de centre et de rayon donné.

En particulier, le dessin des segments à partir du bord du cercle, s'il est bien maîtrisé par l'élève, sera un atout indéniable pour le dessin de la montagne dans l'activité suivante.

## 4. Conduite de l'activité

L'ensemble des programmes ont été construits en vue d'être réutilisés dans chacune des fiches d'activité concernant la programmation graphique sous la TI-83 Premium CE.

Le premier programme sera très souvent réemployé car il permet de « nettoyer » l'écran graphique et de partir d'une page blanche. Tous les programmes ayant trait au dessin commencent par la lettre « D ». Ainsi ce programme d'initialisation de l'écran graphique portera le nom de « DInit ». Il ne comporte pas de difficulté particulière qu'une lecture attentive ne permettrait de dépasser. A noter qu'aurait pu être utilisée la fonction « StoreGDB » qui, couplée à la fonction « RecallGDB », aurait permis de remettre, dans l'état précédant l'appel à DInit, tous les paramètres graphiques de la machine.



Le deuxième programme a pour but de dépasser les enjeux techniques décrits précédemment.

Il doit s'agir là d'offrir, à l'élève, un espace de liberté complet sur l'écran graphique qu'il souhaite construire. L'essentiel est que l'ensemble des outils (primitives) graphiques soit utilisé y compris les fonctions « shade » et « circle » même si celles-ci seront par la suite délaissées soit pour des raisons de vitesse pour la première ou fonctionnelle pour la seconde.

Si, bien sûr, une lecture attentive du listing du programme DMire.8XP est indispensable et si toutes les lignes ne seront pas commentées, on attire l'attention sur les quelques points qui suivent.

### Caractère numérique du paramètre couleur

Pour tracer les rectangles de couleurs en bas à droite de l'écran, on a recours aux lignes

```
For(I,10,24)
For(J,1,10)
Line(10*I+J,5,10*I+J,50,1,I)
End
End
```

La 1<sup>re</sup> boucle concerne les couleurs numérotées de 10 à 24.

La 2<sup>e</sup> boucle concerne la largeur de 10 pixels des rectangles.

Le paramètre 1 permet de tracer des lignes fines de couleurs I.

On trace donc les rectangles sur l'écran graphique de l'abscisse 101 à 250.

Les rectangles ont une longueur de 45 pixels.

### A propos des fonctions et de la fonction Shade

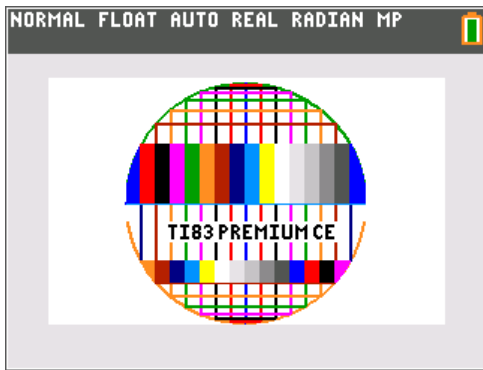
La fonction « shade » est une primitive très puissante de dessin. Malheureusement, dans le contexte qui nous intéresse, nous renoncerons à son utilisation, au profit d'un tracé à l'aide de la fonction « Line » (voir le programme DMire2.8XP).

Observons le tracé de la zone « Jaune » en haut à droite de l'écran.

```
Shade(max(Y2,Y3),Y4,Xmax*3/4,Xmax,4,4,19)
```

On demande à la fonction de hachurer la zone comprise entre le maximum des fonctions  $Y_2$  et  $Y_3$  (représentées initialement en rouge et noir) et la fonction  $Y_4$  (la ligne horizontale orange en haut de l'écran graphique), le tout à l'aide de la couleur 19 c'est-à-dire jaune. Les paramètres 4 et 4 sont des paramètres de motifs du hachurage. Ce travail est effectué dans le dernier quart de l'écran, c'est-à-dire pour des valeurs comprises entre les  $\frac{3}{4}$  de  $X_{max}$  et  $X_{max}$ .

On remarquera que lors de l'appel de la fonction « Shade », celle-ci parcourt l'ensemble des abscisses (ici de 1 à 265) et ce, même si le travail ne s'effectue que sur un intervalle réduit. C'est pour cette raison que le hachurage des 4 zones supérieures de l'écran prend autant de temps. Le recours à la fonction Line s'avèrera plus rapide pour notre objectif de dessin de montagne, même s'il nécessitera davantage de lignes de code.



Pour finir, dans ce troisième programme (DMire2.8XP), c'est sur l'enjeu mathématique que nous nous concentrerons.

Bien sûr, nous pourrions avoir recours à la fonction « circle » qui offre de bonnes performances de tracé.

Mais dans ce cas, nous perdons la maîtrise des coordonnées de l'ensemble des points du cercle. Pour pouvoir effectuer la représentation graphique ci-contre, il faut en passer par la caractérisation « fonctionnelle » du cercle, connaissant son centre et son rayon.

Ce qui donne les lignes de code suivantes :

```
80 → R
132 → C
82 → D
"D+√(R²-(X-C)²) " → Y₅
"D-√(R²-(X-C)²) " → Y₆
FnOn 5
FnOn 6
```

On construit le cercle de centre (132;82) et de rayon 80.

On stocke dans Y₅ la partie supérieure du cercle et dans Y₆ la partie inférieure, fonctions que l'on trace ensuite.

A l'aide du produit scalaire ou du théorème de Pythagore, on obtient la caractérisation analytique du cercle de centre (c ; d) et de rayon R, du type  $(x - c)^2 + (y - d)^2 = R^2$ .

Cela peut être un point de départ pour redécouvrir les fonctions Y₅ et Y₆ par exemple.

Pour finir, on observera la façon de tracer un segment vertical allant d'un bord « supérieur » à un bord « inférieur » du cercle :

```
Line(C+I,Y₅(C+I),C+I,Y₆(C+I),J)
```

On trace un segment vertical d'abscisse C+I et de couleur J.

Pour l'ordonnée, on choisit de partir du périmètre du cercle. On récupère alors les valeurs des fonctions Y₅ et Y₆ pour une abscisse valant C+I, ce qui donne les appels suivants Y₅(C+I) et Y₆(C+I).