

UN DÉCOR POUR NOTRE JEU

Auteur : Jean-Baptiste Civet

TI-83 Premium CE

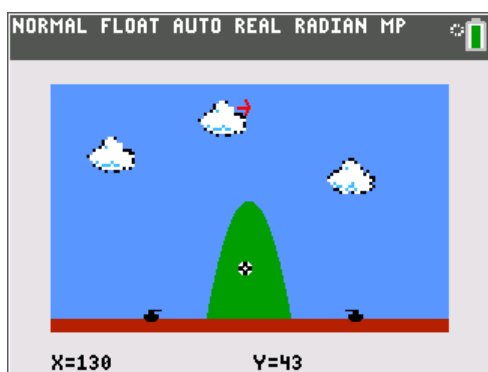
Mots-clés : représentation graphique, programmation graphique.

Fichiers associés : JeuDeTir_fiche2_decor_eleve.pdf, DINIT.8XP, DMONTAGN.8XP, DSPLASH.8XP, DTANK.8XP, DVENT.8XP

1. Objectifs

- Construire le décor de notre jeu, en utilisant les primitives graphiques spécifiques à la TI-83 Premium CE dans l'environnement de programmation de la calculatrice.
- Le décor sera construit à l'aide d'une fonction du second degré et de la connaissance de ses propriétés.
- Ce projet sera l'occasion de comprendre la différence entre les images « background » et le travail sur l'écran graphique en premier plan.
- Il s'agit de la deuxième activité d'un ensemble de trois activités aboutissant à la programmation d'un jeu fonctionnel.

2. Énoncé



Maintenant que les primitives de tracé sont maîtrisées, nous allons mettre à profit nos connaissances techniques et mathématiques pour réaliser un jeu de tir.

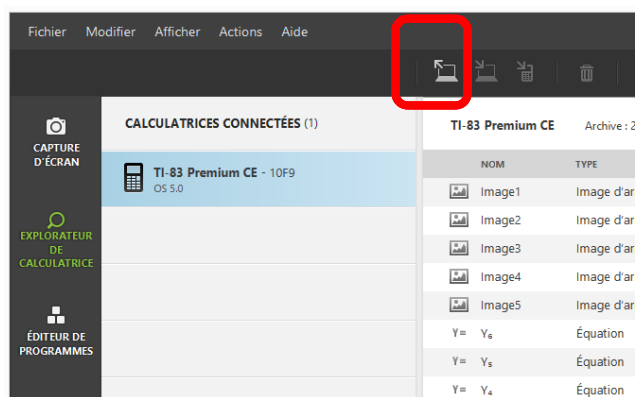
Dans cette deuxième activité, nous préparons les éléments de décor de notre jeu. Les éléments d'interaction seront mis en place dans la dernière activité.

Il s'agit de dessiner la montagne, les tanks, la force du vent et le ciel.

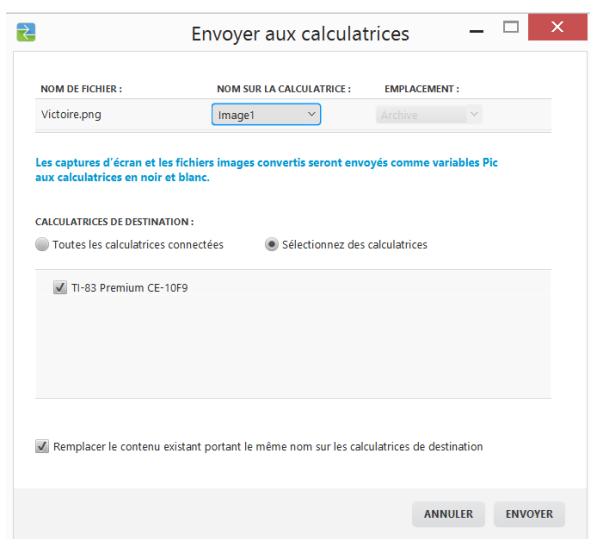
3. Commentaires

Ce jeu est un grand classique de la programmation et ne se démode pas, preuve en est avec le succès, de nos jours, de la saga des « Méchants Oiseaux », cette saga mettant à profit l'interface tactile des appareils. L'enseignant ayant fait ses armes en basic, se souviendra peut être de ce gorille dansant sur le haut d'un building et qu'il fallait atteindre à l'aide d'un boomerang. L'arrivée de la couleur sur la TI-83 était l'occasion d'implémenter de nouveau un tel jeu.

Pour transférer les images de fond (ici le ciel bleu et ses nuages fabriqués par l'élève), il sera nécessaire d'utiliser le logiciel TI-Connect 5.0 ou ultérieur. L'image sera fabriquée à l'aide d'un logiciel de dessin au format « png » et transférée sur la calculatrice, le logiciel TI-Connect se chargeant de la conversion.



On clique sur l'icône « importer depuis l'ordinateur ».



On affecte alors l'image que l'on souhaite importer à l'une des 10 variables disponibles (Image0 à Image9).

Pour démarrer la programmation de cette activité, il faut préalablement résoudre les questions mathématiques suivantes :

- Quelle doit-être la forme générale de la fonction pour que la montagne, de hauteur et largeur variables, soit centrée sur l'écran ? L'usage de la forme canonique d'une fonction du second degré sera fort utile.
- Connaissant l'équation de la parabole, comment déterminer l'espace disponible au sol pour positionner notre premier tank ? Le deuxième tank, par symétrie, disposera du même espace.

4. Conduite de l'activité



Comme toujours, la lecture attentive des programmes est indispensable.

Afin de pouvoir faire communiquer correctement les programmes entre eux, le respect des variables « réservées » est indispensable.

Vu l'ampleur de ce projet en deux parties, il est difficile de suggérer un scénario idéal tant les paramètres sont nombreux (contexte de travail en classe, en TPE, dans le cadre de l'ISN ou bien, selon le moment de l'année, outils mathématiques disponibles auprès des élèves).

Il appartient à l'enseignant d'élaborer son scénario et de prendre la décision de mettre tout ou partie des programmes à disposition des élèves. Une possibilité peut être de laisser les élèves mener à bien la construction de la montagne selon les contraintes données et de se focaliser sur l'aspect mathématique du positionnement des chars d'assaut.

Dessin de la montagne

La montagne est une parabole orientée « sommet » vers le haut et centrée sur le milieu de l'écran. On souhaite faire varier, à chaque nouvelle partie, la hauteur et la largeur de cette montagne.

Connaissant les propriétés de la forme canonique des fonctions du 2nd degré ($f(x) = a(x - \alpha)^2 + \beta$), on propose 132 pour α et H pour β (H la hauteur aléatoire de la montagne). Concernant la largeur de la montagne, on joue sur le paramètre « a » de la forme canonique en identifiant L , la largeur aléatoire de la montagne (comprise entre 0,125 et 0,5). Compte tenu de l'orientation désirée, on saisit donc comme définition :

$$\ll -L(X-132)+H \rightarrow Y_1 \gg$$

Finalement, on propose le code suivant pour le dessin de la montagne :

```
"-L(X-132)^2+H"->Y1
prgmDINIT
CoordOn
BackgroundOn 0
ClrHome
Output(4,4,"INITIALISATION")
60+int(100rand)->H
(12.5+int(38rand))/100->L
For(X,1,265)
If Y1>10
Ligne(X,9,X,Y1,GREEN)
End
For(X,1,9)
Horizontal X,BROWN
End
```

On définit le profil de la montagne dans Y_1 .

On nettoie l'écran de dessin à l'aide du programme créé dans l'activité 1.

On affiche le ciel bleu enregistré dans « image0 ».

On procède ensuite au tirage de la hauteur (dans [60 ; 160]) et de la largeur (dans [0,25 ; 0,5]) de la montagne.

A l'aide d'une boucle, on remplit la montagne sur le même principe que celui qui nous a permis de tracer la mire en suivant le périmètre du cercle dans l'activité 1.

On trace ensuite le sol à l'aide de lignes horizontales marron sur une hauteur de 9 pixels.

On remarquera deux points techniques importants :

- Comme annoncé dans la fiche d'activité 1, on n'a pas recours à la fonction Shade pour dessiner la montagne.
- En incrémentant la variable X dans la boucle, il n'est pas nécessaire de faire appel à $Y_1(X)$ pour obtenir l'ordonnée, seul l'appel à Y_1 suffit.

Positionnement des chars d'assaut

Pour positionner les chars d'assaut, il faut résoudre l'équation $Y_1(X) = 9$. Nous obtenons 2 solutions et nous retiendrons la plus petite pour positionner le 1^{er} char d'assaut. Le 2^e se positionnera par symétrie.

Pour déterminer la place disponible, il faudra également tenir compte de « l'encombrement » du dessin du char d'assaut (évalué à 13 pixels) ainsi que la distance à partir du bord de l'écran à partir de laquelle on souhaite positionner le char (ici 1 pixel).

Ce qui donne, par exemple, le code suivant pour le positionnement des chars d'assaut :

```
132-√((H-9)/L)-13→P
1+int(Prand)→X
1→J
X→U
prgmDTANK
255-int(Prand)→X
-1→J
X→V
prgmDTANK
```

P représente le nombre de pixels disponibles pour positionner les chars à l'ordonnée 9 (niveau du sol), dans la première moitié de l'écran, en tenant compte de la largeur du tank.

Le 1^{er} tank aura la valeur 1 tandis que le 2^{eme} aura la valeur -1 stockée dans J.

On sauvegarde bien les positions des tanks respectivement en U et V.

Le programme DTank dessine le char d'assaut en tenant compte de l'orientation de celui-ci.

Pour finir, la variable J sert tout au long du projet à déterminer l'orientation/le sens des différents dessins/tracés. Par exemple, pour dessiner le tank correctement orienté, voici l'extrait du code :

```
If J=1
Then
Line(X+5,15,X+13,15,BLACK,1)
Else
Line(X,15,X+8,15,BLACK,1)
End
```

Si l'on dessine le joueur de gauche, on oriente le canon vers la droite sinon l'inverse.

X représente la position du coin en bas à gauche du rectangle contenant le tank. Ce rectangle a une largeur de 13 pixels et une hauteur de 7 pixels.

On démarre le tracé à une hauteur de 9.

Représentation du vent

Ce programme n'amène aucun commentaire particulier.

La vitesse du vent est une valeur numérique aléatoire comprise entre -5 et 5.

On considère que le vent souffle horizontalement. La direction du vent sera donnée par le signe de sa vitesse. De gauche à droite si positive et de droite à gauche sinon. Un exemple de programme est donné. Les élèves seront capables de procéder au tirage de la valeur stockée dans E et de tracer la flèche correspondante sur l'écran.

Là encore, comme pour le dessin de la montagne, il peut être intéressant de laisser la pleine autonomie aux élèves. Ces derniers « rebrancheront » alors leur propre programme à l'ensemble des programmes du jeu.

C'est la raison notamment de la multiplication des sous programmes dans ce projet.

Une première mise en musique

Nous pouvons tester l'ensemble de nos procédures. Elles seront réutilisées en l'état dans notre moteur de jeu.

```
prgmDSPLASH
Pause
prgmDMONTAGN
132-√((H-9)/L)-13→P
1+int(Prand)→X
1→J
X→U
prgmDTANK
255-int(Prand)→X
-1→J
X→V
prgmDTANK
prgmDVENT
```

On lance l'écran d'introduction.
On attend l'appui sur la touche « Entrer »
On dessine la montagne.
On calcule la place disponible.
On positionne le premier char d'assaut.
On définit qu'il s'agit du joueur de gauche.
On mémorise sa position (variable réservée U).
On le dessine.
On positionne le deuxième char d'assaut.
On définit qu'il s'agit du joueur de droite.
On mémorise sa position (variable réservée V).
On le dessine.
On calcule et dessine la vitesse du vent.

5. ANNEXE

En vue de la poursuite vers l'activité 3, respecter les affectations de variables :

A : **A**ngle de tir sélectionné par un joueur

D : **D**épart (position de) du tir

E : vitesse du vent (valeur comprise entre -5 et 5) (la variable V est déjà attribuée)

F : **F**orce (vitesse) initiale du tir

G : **G**agné ? (1 si oui, 0 sinon)

H : **H**auteur de la montagne (valeur comprise entre 60 et 160)

J : **J**oueur en action (1 si joueur de gauche, -1 si joueur de droite)

L : **L**argeur de la montagne (valeur, par exemple comprise, entre 0,125 et 0,5)

O : **O**bjectif (position de l')

P : **P**lace disponible

T : **T**emps pour le calcul de la trajectoire du tir

U : Position tank de gauche

V : Position tank de droite

Y1 : équation de la montagne

Respecter de même les noms de programmes suivants :

DInit.8XP : le programme de préparation de l'écran graphique.

DMontagn.8XP : le programme de dessin de la montagne.

DSplash.8XP : le programme de dessin de l'écran d'introduction du jeu.

DTank.8XP : le programme de dessin des chars d'assaut.

DVent.8XP : le programme de dessin de la force du vent.