

Faites vos jeux

Tirage du Loto

Programme pour TI 57
Auteur Jacques Deconchat
Copyright l'Ordinateur de poche
et l'auteur

00	00	0
01	32 2	STO 2
02	32 3	STO 3
03	32 4	STO 4
04	32 5	STO 5
05	32 6	STO 6
06	32 7	STO 7
07	6	6
08	32 0	STO 0
09	86 1	2nd Lbl 1
10	33 1	RCL 1
11	-18	2nd Inv Log
12	-49	2nd Inv Int
13	32 1	STO 1
14	55	x
15	04	4
16	09	9
17	75	+
18	01	1
19	85	=
20	49	2nd Int
21	32 7	STO 7
22	51 2	GTO 2
23	86 3	2nd Lbl 3
24	36	2nd Pause
25	56	2nd Dsz
26	51 1	GTO 1
27	81	R/S
28	71	RST
29	86 2	2nd Lbl 2
30	38 2	2nd Exc 2
31	66	2nd x=t
32	51 1	GTO 1
33	38 3	2nd Exc 3
34	66	2nd x=t
35	51 1	GTO 1
36	38 4	2nd Exc 4
37	66	2nd x=t
38	51 1	GTO 1
39	38 5	2nd Exc 5
40	66	2nd x=t
41	51 1	GTO 1
42	38 6	2nd Exc 6
43	66	2nd x=t
44	51 1	GTO 1
45	22	x≤t
46	51 3	GTO 3

Les programmes de jeux utilisent très souvent des générateurs de nombres pseudo-aléatoires. Voyons comment mettre en œuvre ces derniers en fonction de différents types d'aléas recherchés.

■ Le générateur que nous avons retenu est loin d'être parfait, mais il a le mérite d'être très court. Sur la TI 57, il s'écrit : RCL 1 2nd INV Log 2nd INV Int STO 1. Rappelons qu'il permet, partant d'un nombre initial (noyau) de la forme 0,xxxx d'en obtenir un autre qui servira à fabriquer les nombres désirés et sera à son tour utilisé comme noyau pour le prochain tirage.

En règle générale, il est intéressant de placer ce générateur en sous-programme, on y gagne en souplesse d'utilisation : le générateur fonctionne alors de la même façon que ceux qui sont utilisés en Basic, et qui sont appelés par la fonction RND.

Le tirage d'un nombre entier de 1 à 6 inclus, qui correspond au jet d'un dé, s'effectue ainsi grâce à la séquence SBR 1 × 6 + 1 = 2nd Int, le sous-programme appelé par SBR 1 n'occupant que 6 pas : 2nd Lbl 1 RCL 1 2nd INV Log 2nd INV Int STO 1 INV SBR. Il devient, dès lors, très facile de tirer plusieurs entiers en appelant le sous-programme autant de fois que nécessaire.

Dans un certain nombre de cas, il arrive que l'on ait à faire des tirages

Exemple d'exécution

Taper : 0, 2546 STO 1 RST, puis
R/S, on obtient : 40 14 43 12 34 38
R/S, on obtient : 42 1 2 5 13 38
R/S, on obtient : 37 23 45 5 12 35
etc.



particuliers (tirage pair ou impair, tirage d'un nombre positif ou négatif, par exemple pour des jeux de déplacement...). On peut alors avoir recours à certaines astuces, dont l'arrondi automatique : si l'on convient de représenter un tirage impair par 1 et un tirage pair par 0, on utilisera le très court programme suivant : RCL 1 2nd INV Log 2nd INV Int STO 1 2nd Fix 0 R/S RST. On doit alors prêter attention à la façon dont s'effectue l'arrondi : sur TI 57, il n'affecte que le registre d'affichage. Si l'on a besoin d'utiliser les 0 ou les 1 qui apparaissent à l'affichage, il faudra effectuer un véritable arrondi. Sur les calculatrices TI, il existe une séquence qui effectue cette troncature : c'est la séquence EE INV EE. En incorporant cette

Poker aléatoire

Programme pour TI 57
Auteur Jacques Deconchat
Copyright l'Ordinateur de poche
et l'auteur

00	05	5
01	32 5	STO 5
02	86 6	2nd Lbl 6
03	33 6	RCL 6
04	-18	2nd Inv Log
05	-49	2nd Inv Int
06	32 6	STO 6
07	55	x
08	07	7
09	85	=
10	49	2nd Int
11	32 0	STO 0
12	86 1	2nd Lbl 1
13	33 1	RCL 1
14	38 2	2nd Exc 2
15	38 3	2nd Exc 3
16	38 4	2nd Exc 4
17	32 1	STO 1
18	56	2nd Dsz
19	51 1	GTO 1
20	45	:
21	01	1
22	00	0
23	65	-
24	49	2nd Int
25	32 1	STO 1
26	85	=
27	55	x
28	01	1
29	00	0
30	65	-
31	-49	2nd Inv Int
32	34 1	SUM 1
33	85	=
34	66	2nd x=t
35	51 6	GTO 6
36	75	+
37	33 1	RCL 1
38	-49	2nd Inv Int
39	85	=
40	36	2nd Pause
41	01	1
42	-34 5	Inv SUM 5
43	33 5	RCL 5
44	-66	2nd Inv x=t
45	51 6	GTO 6
46	81	R/S
47	71	RST



séquence dans le programme initial, on obtient : RCL 1 2nd INV Log 2nd INV Int STO 1 EE 2nd Fix 0 INV EE R/S RST ; et les nombres qui apparaissent à l'affichage peuvent réellement être utilisés tels quels.

Dans le cas de jeux où l'on doit obtenir des déplacements aléatoires, il se peut que l'on ait besoin de réaliser des tirages arbitrairement positifs ou négatifs. S'il s'agit par exemple de réaliser un déplacement arbitraire entre -5 et +5, bornes comprises, on pourra fabriquer d'abord un entier entre 0 et 10 (bornes comprises), puis effectuer sur cet entier une translation de valeur -5. Le programme devient alors : RCL 1 2nd INV Log 2nd INV Int STO 1 × 11 - 5 = 2nd Int.

La première partie du programme donne un nombre compris entre 0 et 10, et la translation -5 assure que le résultat final sera bien compris entre -5 et +5 (ne pas oublier que la fonction Int, sur TI 57, ne fait que *supprimer* la partie décimale ; elle ne correspond pas à la fonction *partie entière*).

Si l'on veut maintenant tirer une position au hasard dans un plan ou dans l'espace, il faudra adapter le procédé au résultat recherché (et à la place disponible en mémoire). Pour repérer par exemple la position d'un point dans un cube 10 × 10 × 10, on peut utiliser trois nombres entiers x, y et z, compris entre 0 et 9 et rangés dans trois mémoires diffé-

Exemple d'exécution :

```
Initialisation du générateur
0.42161 STO 6
Entrée des valeurs
72184356 STO 1
13524687 STO 2
45216873 STO 3
82167543 STO 4
Entrée des couleurs
0.1 SUM 1
0.2 SUM 2
0.3 SUM 3
0.4 SUM 4
Début avec RST, R/S
6.1
3.3
5.1 } soit une paire de trois
3.4
7.3
R/S (2° tirage)
8.3
7.2
3.1 } soit une paire de huit
6.3
8.2
etc.
```

rentes. Mais il s'agit d'une solution coûteuse à la fois en pas de programme et en mémoires. Mieux vaut, dans la plupart des cas, se contenter d'un entier de trois chiffres avec la convention : 1^{er} chiffre = x, 2^e chiffre = y, 3^e chiffre = z, d'où le programme suivant : RCL 1 2nd INV Log 2nd INV Int STO 1 × 3 2nd INV Log = 2nd Int. Mais on peut également programmer : RCL 1 2nd INV Log 2nd INV Int STO 1 EE 2nd Fix 3 Inv EE.



Faites vos jeux sur TI 57

La seconde solution, plus courte, présente l'avantage d'un affichage assez facile à interpréter (il y a 3 chiffres en permanence, car les zéros sont visibles) et à utiliser (il suffit de faire des multiplications par 10).

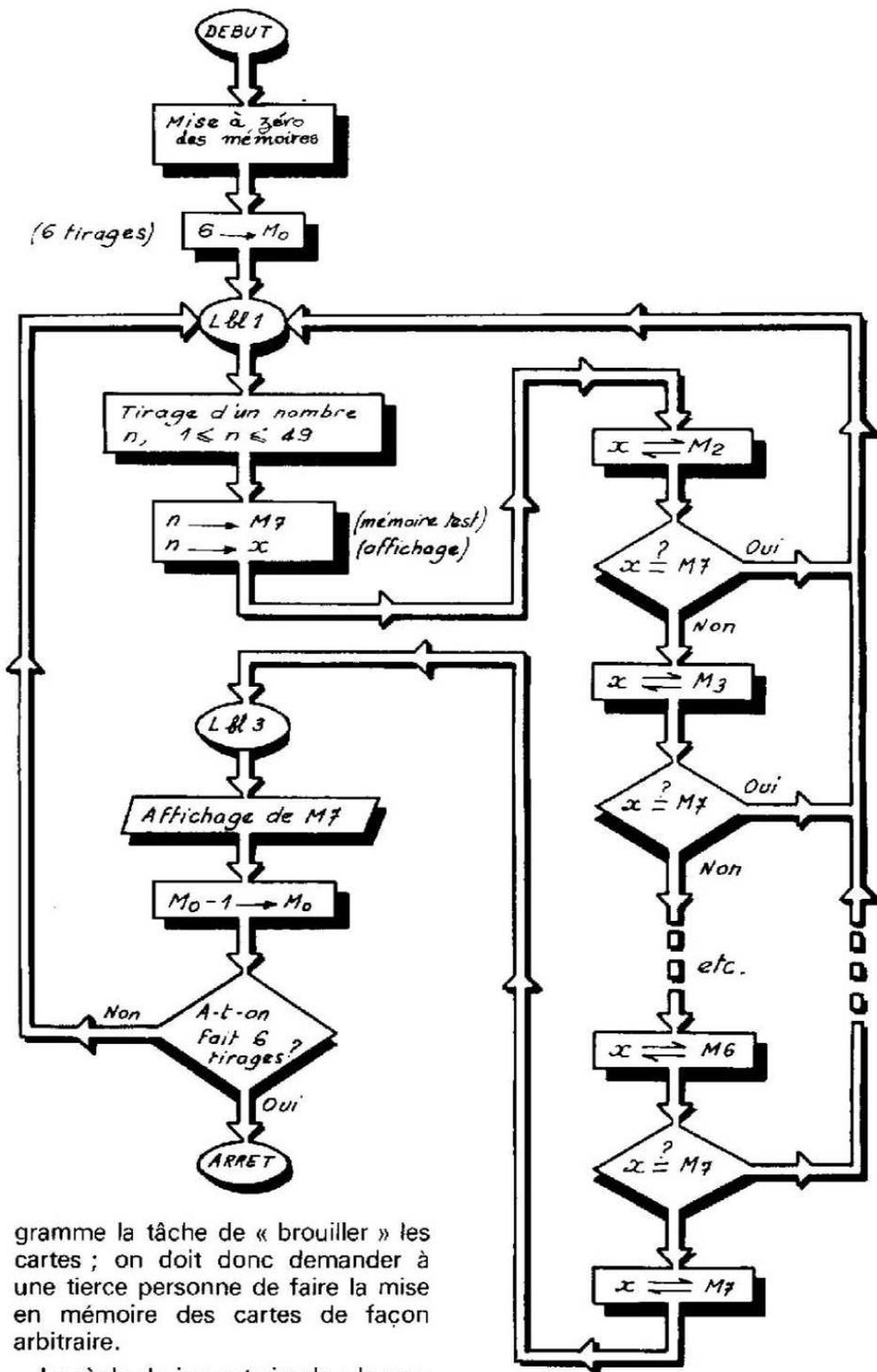
Comment
ne pas tirer deux fois
le même nombre

Bien d'autres problèmes peuvent se poser lors des tirages aléatoires ; l'un des plus délicats à résoudre sur une calculatrice est celui qui consiste à vérifier qu'un nombre n'a pas déjà été tiré (cas du mastermind, du loto, du tirage de cartes, etc.). Les faibles possibilités de mise en mémoire obligent fréquemment à chercher des solutions particulières adaptées au problème à résoudre. Les deux programmes présentés dans cet article (loto et poker) présentent deux solutions différentes à ce problème du tirage sans répétition : l'organigramme du jeu de loto (ci-contre) précise assez bien la structure généralement utilisée.

Avec le loto, il s'agit d'obtenir un tirage de six nombres différents les uns des autres et compris entre 1 et 49. La technique utilisée ici consiste à mettre le dernier nombre tiré dans la mémoire de test et à la comparer, par échange dans les mémoires, avec les tirages précédents : si jamais il y a égalité, on refait un tirage ; sinon, on poursuit les permutations de mémoires.

Le poker aléatoire
32 cartes et
de 2 à 4 joueurs

Le jeu a été conçu pour ressembler autant que possible à un poker normal, mais il n'est cependant pas possible ici de refaire un tirage : la première donne est définitive. Les cartes sont repérées par un numéro, de 1 à 8, et par une couleur, de 1 à 4, ce qui permet toutes les possibilités de distribution d'un jeu de cartes normal. Faute de place, il n'a pas été possible de confier au pro-



gramme la tâche de « brouiller » les cartes ; on doit donc demander à une tierce personne de faire la mise en mémoire des cartes de façon arbitraire.

La règle du jeu est simple : le premier joueur prend la machine et frappe RST puis R/S pour entamer la partie (par la suite, il suffira de frapper R/S). Chaque joueur obtient ainsi cinq affichages consécutifs de la forme x,y où x représente la valeur de la carte (de 1 à 8) et y la couleur. On pourra bien entendu jouer seul contre la machine.

La structure du programme n'est guère compliquée : le nombre de tirages est initialement entré en mémoire 5, puis décompté au fur et à mesure du déroulement de la partie ; on tire un nombre aléatoire qui précise dans quelle mémoire on doit choisir la carte qui est alors extraite,

effacée de la mémoire et affichée avec sa couleur. Si la mémoire désignée est vide, un test conduit à recommencer le tirage.

Afin d'éviter le problème posé par le remplissage initial et arbitraire des quatre mémoires, il serait sans doute intéressant de tirer un nombre compris entre 1 et 32 et de signaler qu'une carte est « sortie » en lui affectant la valeur 0 ou 9 pour éviter d'obtenir deux fois la même carte au cours d'une même partie. Vous pouvez modifier le programme dans ce sens.

□ Jacques Deconchat