

Faisons une petite descente... sur la lune

La trajectoire
d'un vaisseau spatial
obéit à
des lois physiques.
Voyons avec une TI 57
ce qu'il en est
de l'alunissage,
et prenons garde
à la chute des corps !

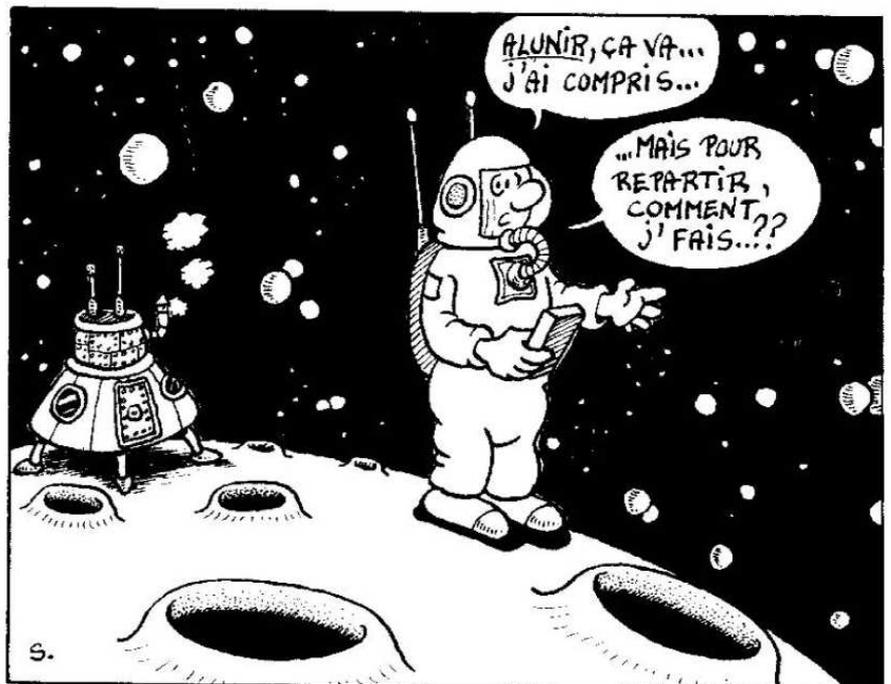
■ Les voyages sur la lune ne sont pas encore à la portée de tout le monde ; ils auraient même tendance à se faire rares ces temps derniers. L'informatique cependant permet de les simuler sans risque d'endommager le matériel ou de mettre en

danger des vies humaines.

Avec un ordinateur de poche, on peut avoir une idée assez bonne de certains aspects de ces voyages. Dans les manuels des micropoches d'ailleurs, on trouve parfois des programmes récréatifs d'alunissage, mais ce n'est pas le cas pour la TI 57. C'est pourquoi je me suis fait ma propre version. Comme la mémoire de cette machine est assez petite, je me suis contenté d'une descente verticale, mais l'alunissage n'en demeure pas moins très délicat tant que l'on n'a pas acquis une bonne expérience de ce type de pilotage.

Le but de chaque partie est bien entendu de se poser en douceur, c'est-à-dire en descendant avec une vitesse qui ne dépasse pas 18 km/h, soit 5 mètres par seconde.

La simulation n'est pas parfaite ;





en particulier, le programme ne tient aucun compte du fait que votre véhicule s'allège lorsqu'il consomme du carburant pour freiner. Mais même ainsi simplifié, l'alunissage reste une manœuvre difficile.

— Comment lutter — — contre — — la pesanteur —

Le programme est construit à partir des lois du mouvement uniformément varié :

$$h = 1/2 \gamma t^2 + V_0 t + h_0$$

$$V = \gamma t + V_0$$

L'accélération (γ) est liée au nombre de litres de carburant utilisés (n) par la relation $\gamma = (n/5) - 1,6$, la valeur 1,6 représentant la pesanteur lunaire. On s'aperçoit ainsi que, dans notre exemple, il suffit d'utiliser 8 litres de carburant pour que γ devienne égal à 0, autrement dit pour que le mouvement devienne uniforme. Si n est plus grand que 8, le mouvement est retardé : votre véhicule freine. Inversement, si n est inférieur à 8, votre vitesse s'accroît.

En choisissant un intervalle de temps de 10 sec, on obtient : $\Delta h = 50\gamma + 10V$ et $\Delta V = 10\gamma$. Pour raccourcir le programme, j'ai posé $\gamma' = 10\gamma$ et les formules deviennent donc :

$$\gamma' = 2n + 16$$

$$\Delta h = 5\gamma' + 10V$$

$$\Delta V = \gamma'$$

Lorsque les calculs effectués par le programme conduisent à une altitude théorique négative, le jeu se termine par le calcul et l'affichage de la vitesse au moment de l'impact (V_i). Cette vitesse est obtenue de la façon suivante : $V^2 - V_0^2 = 2 h' \gamma$ (nota : $h' < 0$) et par conséquent :

$$V_i = \sqrt{V_0^2 - 2 h' \gamma} = \sqrt{V_0^2 - (h' \gamma' / 5)}$$

Pour obtenir un affichage simultané de la vitesse et de l'altitude, on ajoute $h/10^4$ à la valeur absolue de la partie entière de V . Cette solution présente un inconvénient : elle ne permet pas d'afficher le signe de V . Or, en fin de partie, l'engin peut très bien remonter, V étant devenu positif : à force d'utiliser les rétro-fusées pour freiner, on arrive à repartir en marche arrière... Cela dit, il est facile de s'en apercevoir en comparant les altitudes successives.

Au début du jeu, le petit vaisseau spatial est à une altitude de 5000 m. Le pilotage automatique vous rend

les commandes ; vous descendez à une vitesse de 100 mètres à la seconde, et vous disposez de 130 litres de carburant que vous pouvez utiliser comme bon vous semble. Cela devrait normalement suffire.

Après avoir entré le programme, on "fixe" l'affichage à quatre décimales après la virgule : 2nd Fix 4, puis on introduit successivement :

- 130 STO 0 (quantité de carburant, en litres)
- 5000 STO 1 (altitude initiale)
- 100 +/- STO 2 (vitesse initiale).

On peut évidemment modifier chacun de ces trois paramètres en se donnant par exemple davantage de carburant, ou en choisissant une altitude et une vitesse initiales bien inférieures : la descente n'en sera que plus facile.

— Gardez — — du carburant — — pour la fin ! —

Une fois terminée cette initialisation, la descente proprement dite commence avec SBR 1 et l'affichage de "100, 5000" si l'on a retenu les valeurs indiquées plus haut (vitesse initiale de 100 mètres à la seconde et altitude de 5000 mètres). On introduit alors au clavier le nombre de litres de carburant à consommer durant les 10 secondes à venir puis on presse sur R/S. L'affichage indique alors (instruction pause du pas 35) la quantité de carburant restant, puis de nouveau la vitesse en mètres/sec et l'altitude en mètres.

On introduit de nouveau une quantité de carburant avant de presser sur R/S, etc... Le jeu se termine lorsque l'altitude est nulle — quelle que soit la vitesse du vaisseau. Mais on peut considérer que l'alunissage est réussi si la vitesse de l'impact est inférieure ou égale à 5 mètres par seconde.

Pour faire une nouvelle partie, il est inutile de redemander 2nd Fix 4, mais il faut de nouveau indiquer la quantité de carburant (STO 0), l'altitude initiale (STO 1) et la vitesse initiale (STO 2).

En pratiquant ce jeu, vous apprendrez rapidement qu'il est très grave de manquer de carburant (c'est la chute libre), et qu'il ne faut pas freiner plus qu'il n'est nécessaire : lorsque le vaisseau reprend de l'altitude, il "recule" pour mieux... tomber.

□ Jean-Jacques Larousse

EN DOUCEUR...
AUTEUR : JEAN-JACQUES LAROUSSE
COPYRIGHT L'ORDINATEUR
DE POCHE ET L'AUTEUR

0	01	1	
1	00	0	
2	55		MULTIPLIE PAR
3	33	2	RCL 2
4	75		+
5	43		(
6	02		2
7	55		MULTIPLIE PAR
8	33	4	RCL 4
9	65		-
10	01		1
11	06		6
12	44)
13	32	3	STO 3
14	34	2	SUM 2
15	55		MULTIPLIE PAR
16	05		5
17	85		=
18	34	1	SUM 1
19	86	1	2ND LBL 1
20	33	1	RCL 1
21	-76		INV 2ND X >= T ?
22	51	0	GTO 0
23	45		DIVISE PAR
24	04		4
25	-18		INV 2ND LOG
26	75		+
27	33	2	RCL 2
28	40		/X/
29	49		2ND INT
30	85		=
31	81		R/S =
32	32	4	STO 4
33	-34	0	INV SUM 0
34	33	0	RCL 0
35	36		2ND PAUSE
36	71		RST
37	86	0	2ND LBL 0
38	55		MULTIPLIE PAR
39	33	3	RCL 3
40	45		DIVISE PAR
41	05		5
42	84		+/-
43	75		+
44	33	2	RCL 2
45	23		X AU CARRE
46	85		=
47	24		RACINE CARREE DE X
48	49		2ND INT
49	81		R/S