

enseigner les "mathématiques" en classe de cppn

par Gérard Barré
Collège J.B. Pellerin, Beauvais

Je pratique (et à ma demande !) depuis déjà 4 années un sport peu connu des certifiés de Mathématiques : l'enseignement en classe de CPPN (Classe Préprofessionnelle de Niveau). Ces classes recueillent des élèves de 5^e estimés inaptes à suivre en 4^e et n'ayant pas trouvé d'orientation en LEP, ou en CPA, ainsi que des élèves de CM 1 ou CM 2 trop âgés pour pouvoir passer en 6^e, sans parler des redoublants ou retriplants de CPPN (et oui !). Ils y attendent plus ou moins calmement d'avoir 16 ans et de pouvoir goûter aux joies de l'ANPE. Les plus jeunes d'entre eux réussissent parfois à obtenir une entrée en LEP, ou trouvent un patron. Et puis il y a les exclus des autres établissements qui arrivent en cours d'année...

J'ai éprouvé le besoin de formaliser cette pratique en rédigeant quelques réflexions personnelles concernant ces classes et j'y ai adjoint le descriptif d'une activité qui y réussit assez bien.

1. Tentative d'axiomatisation (ni minimale, ni complète...)

Axiome 1 (dit du ghetto) : Les élèves de CPPN sont en situation d'échec scolaire. Durant leur scolarité (ou ce qui en a tenu lieu), une douzaine de collègues (au moins) n'ont pas réussi à les faire rentrer dans le moule. Les mêmes causes produisant souvent les mêmes effets, ce n'est pas avec des méthodes pédagogiques "classiques" qu'il sera possible d'intéresser et de faire progresser ces élèves. De plus, aucun argument de type note/bulletin/orientation/réussite n'a de prise sur eux.

Axiome 2 (dit de l'aventure) : Il n'est pas possible de préjuger de ce qui "marchera" ou "ne marchera pas" dans une classe de CPPN, la réussite ou l'échec d'une activité étant fonction d'un nombre indéterminé de facteurs. Même les "grands succès" tels que : résolution ou fabrication de carrés magiques, jeux inspirés du "Compte est bon", réalisation de dessins à base de figures géométriques, nombres croisés, ... peuvent, sous certaines conditions de température et de pression, être complètement refusés.

Axiome 3 : L'élève de CPPN n'est pas nécessairement sot(te). Certain(e)s sont capables de réflexion/découvertes/pratiques qui feraient pâlir nombre d'élèves dits "normaux" de 4^e ou 3^e.

Axiome 4 : (dit de la variété nécessaire) : L'élève de CPPN n'est pas en mesure de fixer longtemps son attention sur une activité. Quant à espérer pouvoir faire un travail suivi d'une leçon sur la suivante...

Axiome 5 : (dit de l'hétérogénéité totale) : L'effectif et la composition des classes de CPPN sont des plus fluctuants tout au long de l'année scolaire. Evoquer l'hétérogénéité en parlant du niveau de ces élèves est une manière fort élégante de pratiquer l'euphémisme...

Axiome 6 : Toute activité amenant l'élève de CPPN à pratiquer le calcul numérique sous quelque forme que ce soit ne peut être que bénéfique...

Conséquence des axiomes :

a) Toute intervention du type magistral équivaut, en CPPN, à une tentative de suicide pédagogique.

b) L'élève de CPPN n'accepte de travailler que lorsqu'il (elle) est intéressé(e).

c) L'élève de CPPN n'est spontanément intéressé(e) par rien de ce qui a trait aux matières scolaires.

d) L'enseignant en CPPN doit s'habituer à travailler au "coup par coup" et doit toujours être prêt à changer d'activité à tout moment, et ce, peut-être, plusieurs fois au cours d'une même leçon.

e) Ne devraient enseigner en CPPN que des professeurs volontaires.

2. Application pratique :

L'enseignant en CPPN est donc naturellement amené à rechercher des activités de type ludique. Toutes les idées sont bienvenues et la rubrique "Jeux et maths" de la revue de l'APMEP est parfois fort utile... à condition bien sûr de la ramener "au ras des pâquerettes" !

Exemple d'activité connaissant un certain succès en CPPN :

Pratique d'un jeu inspiré de la célèbre et inusable émission de télévision "Le Compte est bon", comme entraînement au calcul numérique.

1°) Pratique collective : La classe est partagée en équipes de trois ou quatre élèves. Un même problème est proposé à tous (pour son élaboration, voir en annexe 1 le programme pour T1 57 LCD). Dès qu'elle croit avoir trouvé, chaque équipe délègue un de ses membres pour écrire, sur la portion de tableau qui lui a été attribuée, sa proposition de solution. Ne sont prises en compte que les solutions intégralement rédigées, et correctement disposées du point de vue mathématique. Dans le cas où deux solutions également valables sont proposées, celle qui l'emporte est celle dont la rédaction est achevée la première. Le professeur joue le rôle d'arbitre dans les cas litigieux. Tout "bon compte" rapporte 2 points. En l'absence de "bon compte", le résultat qui s'en rapproche le plus rapporte 1 point à l'équipe. Je laisse aux élèves un délai de 3 à 4 minutes pour la recherche, et j'interdis à l'équipe qui a délégué un de ses membres au tableau d'intervenir durant la rédaction de la solution (pour la quiétude des classes voisines...).

Cette pratique intéresse les élèves par le côté "compétition" qu'elle engendre. Néanmoins, à vouloir la trop prolonger, elle favorise l'émergence de leaders, d'où découragement de leurs camarades : "ce n'est plus la peine de chercher, ce sont toujours les mêmes qui trouvent". Un autre danger est l'exacerbation de conflits (tensions filles/garçons, racisme,...) souvent latents dans ces classes. D'où :

2°) *Pratique individuelle* : Chaque élève se voit proposer un problème adapté à son niveau de réussite et doit rédiger (toujours de manière mathématiquement correcte) la solution sur sa feuille. Cette fois, les élèves sont prévenus que le "bon compte" existe et aucune contrainte de temps n'est imposée. Pour l'élaboration de tels problèmes voir en annexe 2 le programme pour SHARP PC 1211/1212.

Annexe 1

Programme de génération "Compte est bon" pour TI 57 LCD

Le programme engendre la "cible" et les "plaques". La "cible" est un nombre entier naturel compris entre 1 et 999, les "plaques" sont choisies aléatoirement parmi les nombres : 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 25, 50, 75, 100.

En l'absence de calculatrice, ce programme pourra avantageusement être remplacé par de petits rectangles de bristol sur lesquels on aura écrit les chiffres et les nombres nécessaires, et que l'on tirera au hasard...

Liste du programme :

CIBLE	SBR 0	28 00	00	Appel du ss-pgm générateur de nbres aléatoires		
	3		03 01			
	INV LOG	-	31 02			
	=		95 03			
	INT		58 04			
	LBL 1	23 01	05			
	x ≠ t		51 06			
	0		00 07			
	x ≠ t		51 08			
	INV x = t	-	26 09			
	R/S		13 10			
	PLAQUES	SBR 0	28 00		11	Appel du ss-pgm générateur de nbres aléatoires
		1			01 12	
		5			05 13	
=			95 14			
INT			58 15			
x ≠ t			51 16			
1			01 17			
0			00 18			
x ≥ t			27 19			
GOTO 2		22 02	20			
x ≠ t		51 21	Si nbre aléatoire strictement plus grand que 10,			

	-	75	22	pour 11 affichage de 25,	
	1	01	23	pour 12 affichage de 50,	
	0	00	24	pour 13 affichage de 75,	
	=	95	25	pour 14 affichage de 100.	
	x	65	26		
	2	02	27		
	5	05	28		
	=	95	29		
	GOTO 1	22	01	30	Retour à élimination de 0 et affich.
	LBL 2	23	02	31	Si nbre aléatoire infér. ou égal à 10,
	x \neq 1	51	32		
	GOTO 1	22	01	33	Retour à élimination de 0 et affich.
SOUS-	LBL 0	23	00	34	Sous-progr. de génération d'un nbre
PROGRAMME	RCL 0	71	00	35	(pseudo-aléatoire comp. entre 0 et 1.
	INV LOG	-	31	36	
	FRAC	59	37		
	STO 0	61	00	38	
	x	65	39		
	INV SBR	-	28	40	
CHRONO	LBL 3	23	03	41	
	+	85	42		
	1	01	43		
	=	95	44		
	x \geq 1	27	45	Persistance de l'affich. ss utilisation	
	x $>$ 1	27	46	de pause	
	GOTO 3	22	03	47	

Utilisation du programme :**PART 1**

Avant de commencer : n compris entre 0 et 1 STO 0

Pour obtenir la cible : RST R/S

(En cas d'obtention d'une cible inférieure à 100, refaire RST R/S)

6 pressions successives sur R/S fournissent les 6 plaques.

Pour une autre cible : RST R/S

Pour le chronomètre en secondes (ou presque !) GOTO 3, 0, x \neq 1, 0, R/S

Annexe 2**Programme "Compte est bon" pour sharp PC 1211/1212**

Il n'est, je pense, pas question d'espérer pouvoir rédiger un programme qui, sur PC 1211/1212, permette d'obtenir la solution d'un problème de type "Compte est bon" en moins de quelques heures.

Le programme suivant tourne cette difficulté en fabriquant le problème à partir de sa solution. Je n'ai pas cherché à éviter quelques banalités du genre multiplication ou division par 1 et ne puis garantir que la solution proposé soit la plus simple ou la plus rapide, la seule certitude est qu'elle existe !

Utilisation : Faire RUN, entrer un nombre compris entre 0 et 1 à la demande et... être patient. Si l'imprimante est connectée, la solution du problème s'affiche immédiatement après la question. Pour éviter ce qui peut être considéré comme un inconvénient, il suffit de mettre "STOP" ou "END" à la fin de la ligne 200 et de faire débiter la ligne 300 par l'étiquette "S". SHIFT S en mode DEF donne alors la solution. Sans imprimante, l'affichage se fige sur la question. ENTER fournit la solution.

Programme

```

1 : INPUT "... 0
  <N<1... ?"
  ;N:A(27)=1:
  GOTO 50
3 : "RND" M = N + π : M
  = M ^ 8 + LOG A(2
  7) : N = M - INT M
  ;A(27) = A(27)
  + 1 : RETURN
10 : X = X + Y : RETURN
20 : IF X - Y < = Q LET
  A(6+Z) = 4:
  RETURN
22 : X = X - Y : RETURN
30 : X = X * Y : RETURN
40 : IF X/Y < > INT
  (X/Y) LET A(6
  +Z) = 4 : RETURN
42 : X = X/Y : RETURN
50 : "Z" FOR Z = 1 TO
  6 : GOSUB "RND
  " : O = INT 14N +
  1
60 : IF O < = 10 LET
  A(Z) = O : GOTO
  80
70 : A(Z) = (O - 10) *
  25
80 : NEXT Z
90 : "A" X = 0 : FOR Z
  = 1 TO 6 : A(17 +
  Z) = 0 : A(7 + Z) =
  4 : NEXT Z

```

Description du programme

1 Semence du générateur de nombres aléatoires

3 Sous-programme de génération de nombres pseudo-aléatoires compris entre 0 et 1. La variable A(27) sert à éviter toute périodicité dans la suite de nombres.

10 Sous-programme d'addition.

20 Sous-programme de soustraction et d'élimination des résultats négatifs.

30 Sous-programme de multiplication.

40 Sous-programme de division et d'élimination des résultats non entiers.

50 Fabrication des plaques rangées dans les mémoires A et F.

90 Début de la fabrication de la cible qui sera chargée dans la mémoire X.
Choix des opérations : initialisation à 0 des mémoires H à M et à 4 des mémoires R à W.

95 : GOSUB "RND": O = INT 6N + 1; X = A(O); R = A(O) :A(O) = A(O) + 1 000	95 Chargement dans X et R du contenu de la mémoire A à F (plaque) choisie et augmentation de 1000 du contenu de cette mémoire (pour ne pas l'utiliser deux fois !).
100 : Z = 0	100 Initialisation à 0 de Z, le nbre d'opérations.
105:Z = Z + 1:IF Z = 6 THEN 160	105 Après au plus 6 opérations, fin de cette partie du programme.
110 : GOSUB "RND": A(6 + Z) = INT 4 N	110 Chargement dans les mémoires G à K d'un nombre aléatoire compris entre 0 et 3.
115 : GOSUB "RND": P = INT 6N + 1	115 Choix d'une plaque non encore utilisée.
120 : IF A(P) <= 100 THEN 130	
125 : Q = P < > 6	125 Q est un indicateur binaire permettant de revenir à A après l'examen de F.
126 : P = Q * (P + 1) + 1 - Q:GOTO 120	
130 : Y = A(P):A(P) = A(P) + 1000:A(18 + Z) = Y	130 Chargement dans Y de la plaque choisie et de Y dans une mémoire de R à W.
140 : GOSUB 10 + 10 * A(6 + Z)	140 Appel du sous-programme d'opération en fonction du contenu de la mémoire de G à K: 0 : + 1 : - 2 : x 3 : /
150 : GOTO 105	
160 : FOR Z = 1 TO 6: IF A(Z) > 100 LET A(Z) = A(Z)) - 1000:NEXT Z:GOTO 175	160 Remise des mémoires de A à F dans l'état initial.
170 : NEXT Z	
175 : IF (X < 100) + (X > 999)BEEP 1 :GOTO "A"	175 Si la cible fabriquée n'est pas un nombre à trois chiffres, faire BIP et recommencer.
200 : PRINT X;"AVE C- ->":PRINT A;B;C;D;E;F	200 Affichage du problème.
300 : PRINT "SOLUT ION - ->":O = R	300 Affichage de la solution proposée.
310 : FOR Z = 1 TO 5	
320 : IF A(6 + Z) = 4 NEXT Z:GOTO 400	

```
330 : IF A(6+Z)=0
      LET Y$ = "+" : Q
      = 0 + A(18+Z) :
      GOTO 350
335 : IF A(6+Z)=1
      LET Y$ = "-" : Q
      = 0 - A(18+Z) :
      GOTO 350
340 : IF A(6+Z)=2
      LET Y$ = "*" : Q
      = 0 * A(18+Z) :
      GOTO 350
345 : Y$ = "/" : Q = O / A
      (18+Z)
350 : P = 18+Z : PRINT
      O ; Y$ ; A(P) ; " =
      " ; Q
365 : O = Q
370 : NEXT Z
400 : PRINT " " :
      GOTO "Z"
```