Probabilités, simulation et algorithmique (pour TI)

Extrait du programme officiel :

Dans la partie Statistiques et probabilités :

Réalisation d'une simulation.	Concevoir, mettre en œuvre et exploiter des simulations de situations concrètes à l'aide du tableur ou d'une calculatrice.	À l'occasion de la mise en place d'une simulation, on peut : utiliser les fonctions logiques d'un tableur ou d'une calculatrice, mettre en place des instructions conditionnelles dans un algorithme.

Dans la partie algorithmique :

Les élèves, dans le cadre d'une résolution de problèmes, doivent être capables :

- de programmer un calcul itératif, le nombre d'itérations étant donné ;
- de programmer une instruction conditionnelle, un calcul itératif, avec une fin de boucle conditionnelle.

Description du document

Ce document n'a pas été rédigé comme une activité élève.

Il s'agit d'une suite d'exemples et d'exercices, parfois commentés, pouvant être donnés à des élèves (principalement en seconde mais aussi en première ou terminale pour certains) à différents moments de l'année dans le cadre de la simulation en statistique, leur dénominateur commun étant l'utilisation d'une simulation réalisée à l'aide d'une calculatrice.

Motivation :

Ce type d'activités permet de faire travailler deux aspects du programme : l'algorithmique et la familiarisation avec l'aléatoire. Les résultats obtenus fournissent des données qu'il est possible d'exploiter de diverses manières afin de réinvestir les notions rencontrées lors du chapitre de statistiques descriptives. Créer des algorithmes pour simuler des expériences aléatoires est une activité motivante pour les élèves et appréciée de beaucoup d'entre eux.

Remarque :

Pour réaliser ce type d'activité en classe (et surtout pour le corriger), il est important que tous les élèves aient le même type de calculatrice. C'est pourquoi il est préconisé que l'équipe des enseignants de l'établissement se mette d'accord pour demander aux élèves d'avoir une marque déterminée de calculatrice.

Pour l'écriture de ce document, nous avons fait le choix de l'utilisation d'une calculatrice de la marque TEXAS INSTRUMENT ; elle peut bien sûr être remplacée par une calculatrice d'une autre marque en adaptant éventuellement les questions.

Certains programmes ont été écrits en français et d'autres en anglais, les deux langues pouvant être employées sur calculatrice.

Il faut aller à la ligne après les instructions « then » (alors) et « else » (sinon). Cela n'a pas été fait sur ce document pour gagner de la place. De même certaines affectations ont été regroupées sur une même ligne.

I. Comment simuler une expérience aléatoire.

NbrAleat (ou **rand** en anglais) simule une variable aléatoire qui suit la loi uniforme sur l'intervalle [0 ;1[.

EntAleat(n1,n2) (**randInt(n1,n2)** en anglais) simule une variable aléatoire discrète et uniforme sur les entiers de l'intervalle [n1 ;n2].

Ces instructions se trouvent dans le sous menu PRB du menu MATH.

Les instructions nécessaires pour les tests ($=, \neq, <, \leq \cdots$) se trouvent, avec les connecteurs logiques, dans un des menus TEST.

Les instructions de programmation se trouvent dans le menu PGRM.

Dans la suite du document, le symbole → a été utilisé pour l'affectation d'une valeur à une variable ($\boldsymbol{sto} \rightarrow$ sur la calculatrice).

Lancers de pièces.

```
Si EntAleat(1,2)=1
Alors Disp « Face »
Sinon Disp « Pile »
Fin
```

```
Si NbrAleat <0,5
Alors Disp « Face »
Sinon Disp « Pile »
Fin
```

```
Partent (2*NbrAleat)→D
Si D=1
Alors Disp « Face »
Sinon Disp « pile »
Fin
```

On peut bien sûr adopter la convention Pile =1 Face =0. Le deuxième programme peut être plus facilement modifié pour simuler le lancer d'une pièce non équilibrée ou toute expérience à deux issues. Le premier et le troisième pouvant eux être transformés pour simuler le lancer d'un dé équilibré.

Tirages dans une urne contenant 5 boules rouges, 3 boules blanches et 2 noires.

```
NbrAleat→ X
Si X<0.5
Alors Disp « rouge »
Sinon   Si X<0.8
        Alors Disp « Blanc »
        Sinon Disp « Noir »
         Fin
Fin
```

Une expérience

```
0→ R   0→ B  0→ N
Pour(I,1,4)
        NbrAleat→X
        Si X<0.5
        Alors R+1→ R
        Sinon   Si X<0.8
                Alors B+1→ B

                Sinon N+1→ N
                 Fin
        Fin
Fin
Disp « R= »,R
Disp « B= »,B
Disp « N= »,N
```

4 tirages avec remise

Nombre de garçons dans une famille de 3 enfants.

```
RandInt(0,1)+Randint(0,1)
+Randint(0,1) → N
Disp N
```

simple

```
0 → X
For (I,1,3)
        If Randint(1,2)=1
        Then 1+X→ X
        End
End
Disp X
```

Plus dur…

Combien de fois faut-il lancer un dé avant d'obtenir 6 ?

```
0 → N
0 → X
While X ≠ 1
        N+1→ N
        If Randint(1,6)=6
        Then 1→ X
        End
End
Disp N
```

```
0 → N
0 → X
While X ≠ 1 and N < 1000
        N+1→ N
        If Randint (1,6)=6
        Then 1→ X
        End
End
Disp N
```

La deuxième version pour ceux qui peuvent trouver gênant les univers infinis.

II. Utiliser une simulation pour conjecturer.

Problème 1 : Comme un enfant sur deux est une fille et un sur deux un garçon, il y a environ autant de familles à quatre enfants avec deux filles et deux garçons que de familles à quatre enfants avec 3 filles et un garçon. Vrai ou Faux.

Première approche :
On effectue la simulation (comme pour le nombre de garçons dans une famille de 3 enfants – voir plus haut).On la relance plusieurs fois pour se faire une idée. En classe, il est possible de le faire une trentaine de fois puis de mutualiser les résultats ce qui nous donne environ un millier d'expériences.

Approfondissement :
On peut faire calculer les fréquences des événements « 2 garçons » et « 1 garçon » sur un millier d'expériences.

```
0→ A
0→ B
For (I,1,1000)
        Randint(0,1)+ Randint(0,1)+ Randint(0,1)+ Randint(0,1)→ X
        If X=1
        Then A+1→ A
        Else    If X=2
                Then B+1→ B
                End
        End
End
Disp A/1000 ,B/1000
```

Suivant les cas, il est possible de :

- donner le programme et de demander aux élève ce qu'il fait,
- donner une version incomplète et le faire compléter,
- donner le programme avec une deux erreurs et le faire corriger.

On peut évidemment prolonger cette activité par un calcul de probabilité.

Deuxième possibilité (utilisation des listes) :
On peut simuler les naissances dans 1000 familles et récupérer le nombre de familles à n garçons $0 \leq n \leq 4$ dans une liste (L2 par exemple).

```
For (I,1,1000)
Randint(0,1)+ Randint(0,1)+ Randint(0,1)+ Randint(0,1)→ X
L2(X+1)+1→ L2(X+1)
End
```

X est le nombre de garçons obtenus à chaque simulation, on ajoute une unité dans l'élément correspondant dans la liste (attention au décalage)

Il faut dans ce cas avoir vidé la liste L2 au préalable.
On peut manuellement mettre les valeurs de 0 à 4 dans la liste L1 pour utiliser ensuite les fonctions statistiques de la calculatrice.

Pour les meilleurs élèves, on peut envisager de leur demander de tout automatiser pour obtenir un programme ressemblant à :

```
ClrList L1, L2
For(J,1,5)
J-1→ L1(J)
0→ L2(J)
End
For (I,1,1000)
Randint(0,1)+ Randint(0,1)+ Randint(0,1)+ Randint(0,1)→ X
L2(X+1)+1→ L2(X+1)
End
```

On commence par mettre des 0 dans les éléments de la liste L2 et 0,1,…4 dans L1.

Le programme met environ 4 minutes pour s'exécuter (TI-83).

Problème 2 : Lorsqu'on lance 200 fois de suite une pièce de monnaie, est-il fréquent d'obtenir 6 fois de suite (ou plus) pile ?

0→ P 0→ N For (I,1,200) Randint(0,1) → X X*N +X →N If N>P Then N → P End End Disp P	1 correspond à Pile N est la longueur de la série de Pile en cours. Le programme met quelques secondes pour s'exécuter.

Un exemple d'exploitation de la situation :
On donne le programme en remplaçant 200 par 20 et en précisant que le « 0 » correspond à FACE et le « 1 » correspond à PILE.

1. On suppose que les nombres obtenus avec l'instruction Randint(0,1) sont successivement :
   0 1 0 0 1 0 1 1 0 1 1 1 1 0 0 1 1 1 0 1 Remplir un tableau d'état des variables X, N et P.
   Qu'affiche le programme ?
2. On suppose cette fois que la série obtenue est :
   1 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0 0 0 1 Qu'affiche le programme ?
3. A quoi correspond la valeur affichée par ce programme ?
4. Programmer votre calculatrice (on remplace 20 par 200). Exécuter dix fois ce programme en notant les résultats obtenus.
5. Par groupe de 4 élèves : Echanger vos résultats de sorte que chacun ait 40 résultats de simulation. Déterminer la moyenne, la médiane, les quartiles de la série obtenue. Représenter cette série.
6. Mona veut parier que sur une série de 200 lancers elle obtiendra une série de six « Pile » consécutif. Acceptez-vous ?

III. Autres exercices

Exercice 1 (1èreS ou ES, 2nde pour les questions 1 à 3) : Pierre joue au jeu suivant. On lance 12 fois de suite une pièce de monnaie équilibrée. Si le nombre de Pile obtenu est supérieur ou égal à 9 il gagne 10 euros, sinon il perd 1 euro. Est-il intéressant pour lui de jouer ?

1. Etablir un programme sur calculatrice qui simule une expérience et affiche le gain (10 ou -1) de Pierre ?
2. Modifier le programme précédent pour qu'il simule dix parties et affiche le gain final de Pierre.
3. Utiliser le programme de la deuxième question pour conseiller Pierre qui se demande s'il doit jouer ou non.
4. Précisez la loi de probabilité du nombre de pile obtenu lorsqu'on lance douze fois un dé.
5. Donner une valeur approchée de la probabilité pour que Pierre gagne 10 € lorsqu'il joue une seule partie et en déduire une valeur approchée de l'espérance mathématique de la variable aléatoire G (gain de Pierre). Que peut-on en conclure ?

Exercice 2 : (Un classique !)
Bernard décide de tenter sa chance à un jeu de Pile ou Face. Il dispose de 10€ qu'il accepte de perdre en totalité. Convaincu que la pièce tombera plus souvent sur Pile que sur Face. Il décide de miser systématiquement sur Pile en adoptant la stratégie suivante :
- A la première partie il mise 1€.
- Lorsqu'il vient de gagner, il mise 1€.
- Lorsqu'il vient de perdre, il double la mise précédente s'il lui reste assez d'argent sinon il s'arrête de jouer.
- Il joue au maximum dix parties.

Construire un algorithme qui simule les parties de Bernard. L'algorithme doit afficher le nombre de parties jouées ainsi que la somme gagnée ou perdue par Bernard.

Exercice 3 :

a) Bernard est certain qu'en lançant 10 fois de suite une pièce de monnaie, à un moment donné le nombre de Pile et le nombre de Face obtenu seront égaux. Ses arguments n'ont pas convaincu Claire.
Construire un algorithme qui simule cette expérience et l'utiliser pour déterminer qui a raison des deux.

b) Bernard est convaincu qu'en lançant 12 fois de suite un dé équilibré il obtiendra presque à coup sûr au moins une fois chacun des 6 résultats possibles. Ses arguments n'ont pas convaincu Claire.
Construire un algorithme qui simule cette expérience et l'utiliser pour déterminer qui de Bernard ou de Claire a raison.

Exercice 4 : Un jeu pas vraiment équilibré (1ère ou terminale)

```
PROGRAM:CHOIX
:randInt(1,8)→X
:If X>3
:Then
:1→A
:Else
:0→A
:End
:Input "A TOI 0 OU 1",B
:3-A-B→G
:If G=2
:Then
:Disp "TU PERDS",G,"EU"
:Else
:Disp "TU GAGNES",G,"EU"
:End
```

1. On suppose que vous jouez au jeu « CHOIX » dont le programme est donné ci-contre. Vous choisissez de manière équiprobable 0 ou 1 (par exemple, en lançant une pièce de monnaie équilibrée).
   a. Préciser les phrases exactes que peut afficher la calculatrice à la fin du programme.
   b. Calculer la probabilité que la calculatrice affiche chacune des 3 phrases possibles.
   c. Déterminer votre espérance de gain.
2. On ne choisit plus 0 ou 1 de manière équiprobable. Démontrer que l'espérance de gain reste négative.