Thème: simulation ; intervalle de fluctuation

Niveau: 2nde

TI-83 Premium CE

python™

Le lièvre et la tortue

F. GIROD

Compétences visées

- **chercher**, expérimenter – en particulier à l'aide d'outils logiciels ;
- **modéliser**, faire une simulation, valider ou invalider un modèle ;
- **représenter**, choisir un cadre (numérique, algébrique, géométrique...), changer de registre ;
- **calculer**, appliquer des techniques et mettre en œuvre des algorithmes ;
- **communiquer** un résultat par oral ou par écrit, expliquer une démarche.

Ces compétences sont mises en œuvre dans le cadre de l'extrait du programme de 2nde GT ci-dessous :

« Simuler N échantillons de taille n d'une expérience aléatoire à deux issues. Si *p* est la probabilité d'une issue et *f* sa fréquence observée dans un échantillon, calculer la proportion des cas où l'écart entre *p* et *f* est inférieur ou égal à $\frac{1}{\sqrt{n}}$ »

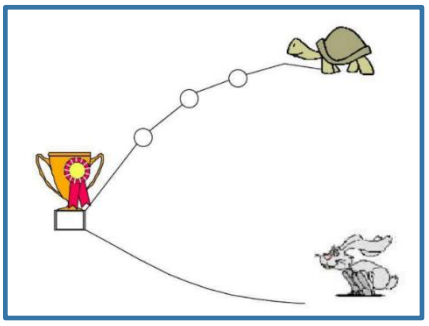
Situation déclenchante

Le lièvre et la tortue :

On lance un dé bien équilibré à six faces :

- à chaque lancer de dé, la tortue avance d'une case si le dé affiche les numéros 1, 2, 3, 4 ou 5 ;
- le lièvre gagne si le dé affiche le numéro 6.

Le but de la partie est d'atteindre la coupe : qui a la situation la plus favorable ?



Problématique

Comment modéliser la situation pour déterminer qui du lièvre ou de la tortue a la situation la plus favorable ?

Thème: simulation, intervalle de fluctuation

Niveau: 2nde

TI-83 Premium CE

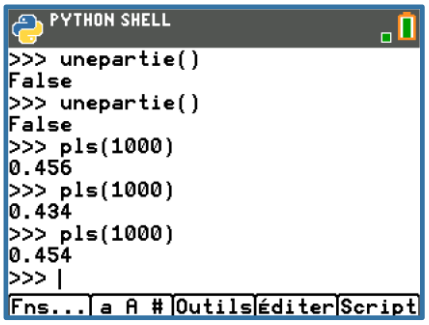
python

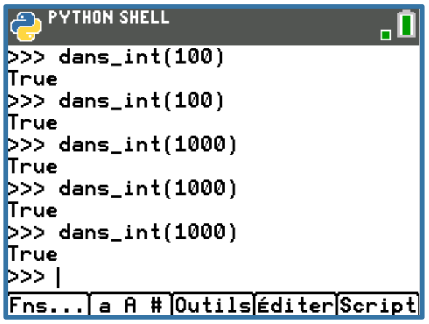
F. GIROD

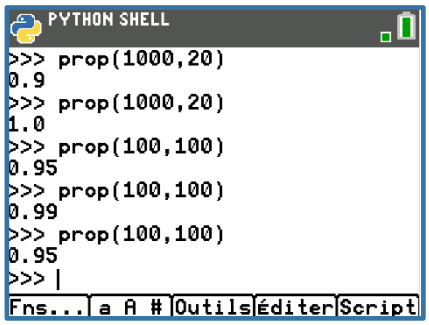
Fiche méthode

Proposition de résolution

- Écrire la fonction **unepartie** (sans argument) simulant une partie ; elle renvoie :
    - True pour une victoire de la tortue ;
    - False pour une victoire du lièvre.
- Écrire la fonction **pls** qui prend pour argument *n*, le nombre de répétitions voulues. Elle renvoie la fréquence de victoires de la tortue pour *n* parties.
- Écrire la fonction **dans_int** qui prend pour paramètre *n*, le nombre de parties simulées. Elle renvoie le booléen True si la fréquence de victoires de la tortue est à une distance inférieure ou égale à $\frac{1}{\sqrt{n}}$ de *p* (la probabilité de victoire de la tortue calculée par ailleurs égale à 0,482). Elle renvoie False si ce n'est pas le cas.
- Écrire la fonction **prop** qui a pour premier paramètre *n*, la taille de l'échantillon, et comme second paramètre *N*, le nombre d'échantillons ; elle renvoie la proportion des *N* échantillons de taille *n* testés qui sont tels que l'écart entre *p* et *f* est inférieur ou égal à $\frac{1}{\sqrt{n}}$.



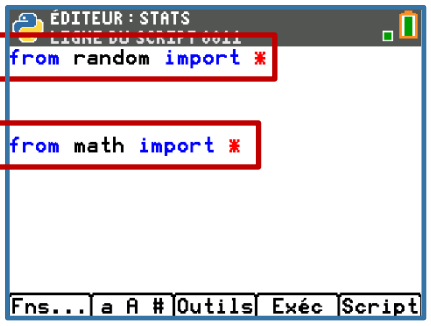




Remarque

Importation en préambule du code de la bibliothèque « random » par « **from random import ✶** » pour pouvoir utiliser la fonction **randint()**

importation de la bibliothèque « math » par « **from math import ✶** » pour pouvoir utiliser la fonction **fabs** (valeur absolue)



Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus

Thème: simulation, intervalle de fluctuation

Niveau: 2nde

TI-83 Premium CE

python

F. GIROD

Fiche méthode

Etapes de résolution

La fonction **unepartie** n'a pas de paramètre et renvoie un booléen (True ou False) selon que la simulation donne une victoire de la tortue ou non.

```
ÉDITEUR : TORTUE
LIGNE DU SCRIPT 0011
from random import*
from math import *

def unepartie():
  for i in range(4):
    if randint(1,6)==6:
      return False
  return True

Fns... | a A # | Outils | Exéc | Script
```

Elle simule au plus près le jeu : on répète jusqu'à quatre lancers de dé à six faces, et dès que la valeur 6 est sortie, **la fonction renvoie False.** En effet, dès que le script rencontre « **return** », la fonction s'arrête ; c'est aussi un gain de temps, les répétitions seront moins nombreuses.

La fonction **pls** prend pour paramètre `n`, le nombre de parties simulées. Elle renvoie la fréquence de parties gagnées par la tortue.

Elle utilise la fonction **unepartie** et un compteur (variable `c`) comptabilise le nombre de parties simulées donnant une victoire de la tortue.

```
ÉDITEUR : TORTUE
LIGNE DU SCRIPT 0023

def pls(n):
  c=0
  for i in range(n):
    if unepartie():
      c+=1
  return c/n

Fns... | a A # | Outils | Exéc | Script
```

C'est l'utilisation successive de ces fonctions qui rend le programme très lisible et efficace.

A noter le raccourci **`c+=1`** qui signifie **`c=c+1`**

La fonction **dans_int** a pour paramètre `n` ; elle renvoie le booléen True ou False selon que la distance entre la fréquence de victoires de la tortue après `n` parties (donnée par **pls**(`n`)) et *p* est inférieure ou égale à $\frac{1}{\sqrt{n}}$ ou pas.

```
ÉDITEUR : TORTUE
LIGNE DU SCRIPT 0030

def dans_int(n):
  return fabs(pls(n)-0.482)<=1/s
      qrt(n)

Fns... | a A # | Outils | Exéc | Script
```

Bien noter que « $|\text{pls}(n) - 0{,}482| \leqslant \frac{1}{\sqrt{n}}$ » est un booléen ; il prend la valeur True si le test est vérifié, la valeur False sinon. Cela permet d'avoir une syntaxe très concise.

La fonction **prop** a pour paramètres `n` (taille d'un échantillon de parties) et *N* (nombre d'échantillons) ; elle renvoie la proportion des *N* échantillons de taille `n` qui ont une fréquence de victoires de la tortue proche de *p* (distance inférieure ou égale à $\frac{1}{\sqrt{n}}$).

```
ÉDITEUR : TORTUE
LIGNE DU SCRIPT 0037

def prop(n,n_rep):
  c=0
  for i in range(n_rep):
    if dans_int(n) :
      c+=1
  return c/n_rep

Fns... | a A # | Outils | Exéc | Script
```

Elle utilise efficacement la fonction **dans_int**.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus

Thème: simulation, intervalle de fluctuation

Niveau: 2nde

TI-83 Premium CE

python

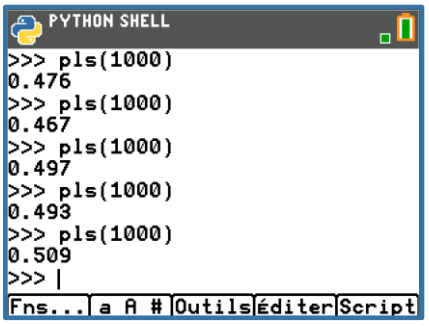
F. GIROD

Fiche méthode

Remarques

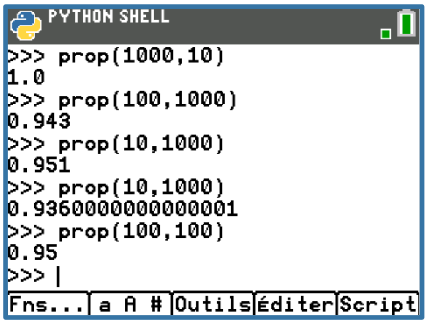
Il peut être intéressant d'exécuter plusieurs fois de suite l'instruction **`pls(1000)`** afin de constater la fluctuation d'échantillonnage sur des échantillons de même taille.

L'appel de la même instruction par la flèche directionnelle vers le haut est particulièrement utile ici : en effet, un appui sur la flèche directionnelle vers le haut appelle l'instruction tapée précédemment (deux appuis successifs rappellent l'instruction écrite deux lignes au-dessus, et ainsi de suite) ; en appuyant sur **`entrer`**, on exécute à nouveau l'instruction écrite dans la console.



L'instruction **prop**(n,N) est à manier avec précaution ! Dans la mesure où **prop**(1000,100) demande 100 répétitions de tests sur des échantillons de taille 1000 ... le calcul prendra une vingtaine de secondes.

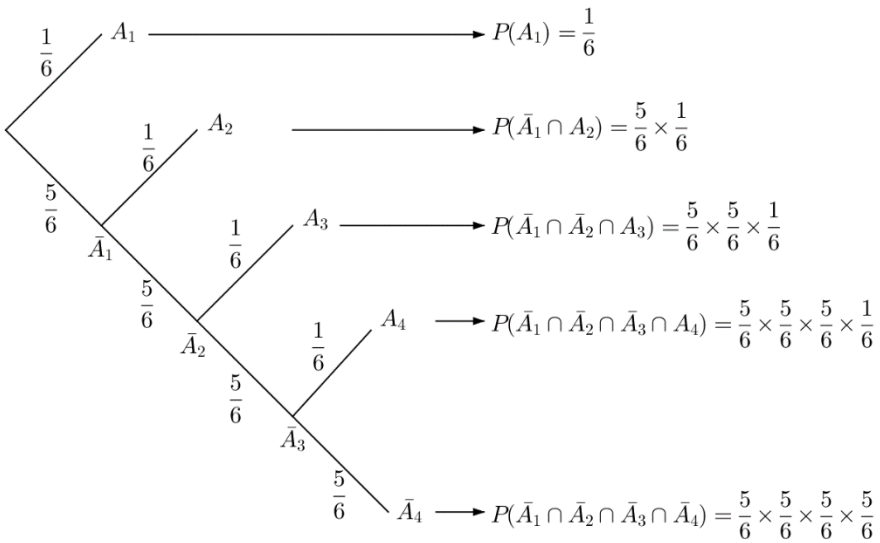
Le calcul de **prop**(1000,1000) est à déconseiller !



Pour aller plus loin

Pour ce qui est du calcul de la probabilité de victoire de la tortue, on peut proposer à un groupe d'élèves d'effectuer cet approfondissement / recherche, par exemple à l'aide d'un arbre comme celui présenté ci-contre :

On obtient ainsi la probabilité de victoire de la tortue : $p = \left(\frac{5}{6}\right)^4 \approx 0{,}482$



on note A_i l'événement « obtenir un numéro 6 au i^{eme} lancer »

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus