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 $\frac{1}{\sqrt{n}}$ »

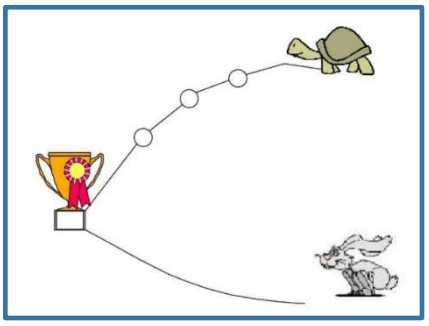
Situation déclenchante

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 on tombe sur les numéros 1, 2, 3, 4 ou 5 ;
- le lièvre gagne s'il on tombe sur le numéro 6.

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

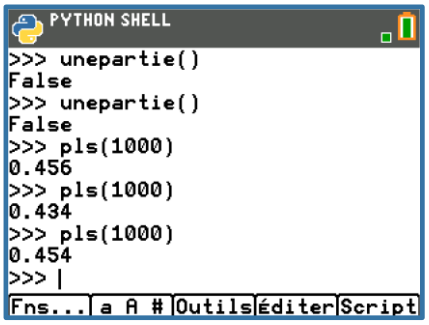
python

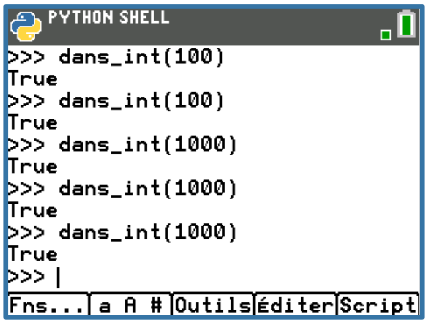
F. GIROD

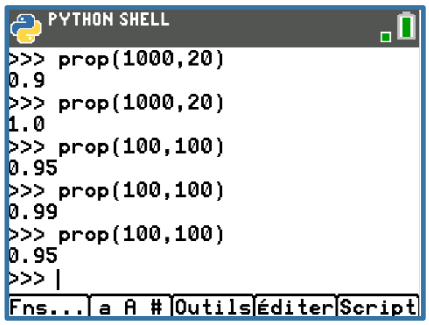
Fiche méthode

Proposition de résolution

- La fonction **unepartie**() simule une partie en donnant le résultat sous forme de booléen :
    - True pour une victoire de la tortue ;
    - False pour une victoire du lièvre.
- La fonction **pls**() permet d'obtenir un échantillon de n parties et renvoie la fréquence de victoires de la tortue. Elle a pour paramètre n, le nombre de répétitions.
- La fonction **dans_int**() a pour paramètre n, la taille de l'échantillon. Elle teste si l'écart entre p et f est inférieur ou égal à $\frac{1}{\sqrt{n}}$, p représentant la probabilité de victoire de la tortue (calculée par ailleurs), et f la fréquence de victoire de la tortue pour cet échantillon. Elle retourne elle aussi un booléen.
- La fonction **prop**(,) qui a pour premier paramètre la taille de l'échantillon (n), et comme second paramètre, le nombre d'échantillons (N) donne 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()** simule une partie entre le lièvre et la tortue

Cette fonction n'a pas de paramètre en entrée et renvoie True pour simuler une victoire de la Tortue et False pour une victoire du Lièvre.

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, « **return » fait sortir de la boucle dès qu'on le rencontre** ; c'est aussi un gain de temps, les répétitions seront moins nombreuses.

```
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
```

la fonction **pls**() renvoie la fréquence de parties gagnées par la tortue.

Elle utilise la fonction **unepartie**() et un compteur (variable c) comptabilise le nombre de réalisations ; elle a pour paramètre n, le nombre de répétitions.

```
ÉDITEUR : TORTUE
LIGNE DU SCRIPT 0015
    if randint(1,6)==6:
      return False
  return True

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 très efficace.

La fonction **dans_int**() renvoie le booléen True ou False selon que la distance entre fréquence de victoire 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 0021
    if unepartie():
      c+=1
  return c/n

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

def prop(n,n_rep):
  c=0
Fns... | a A # | Outils | Exéc | Script
```

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

La fonction **prop**(,) renvoie la proportion des N échantillons de taille n qui ont une fréquence de victoire de la tortue proche de *p* (distance inférieure ou égale à $\frac{1}{\sqrt{n}}$).

```
ÉDITEUR : TORTUE
LIGNE DU SCRIPT 0025

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

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**().

A noter, comme pour la fonction **pls**(), le raccourci `c+=1` qui signifie `c=c+1`

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

Thème: simulation, intervalle de fluctuation

Niveau: 2nde

TI-83 Premium CE

python

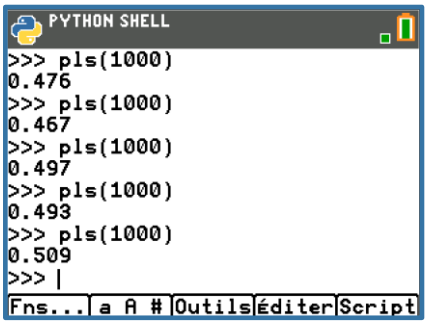
Fiche méthode

F. GIROD

Remarques

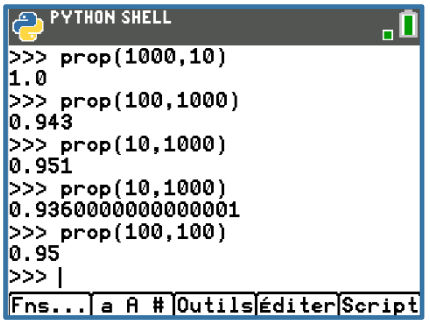
Il peut être intéressant de constater la fluctuation d'échantillonnage dès la fonction **pls**() faite en l'exécutant plusieurs fois de suite avec la même valeur pour le paramètre n.

L'appel de la même instruction par la flèche directionnelle vers le haut est particulièrement utile ici.

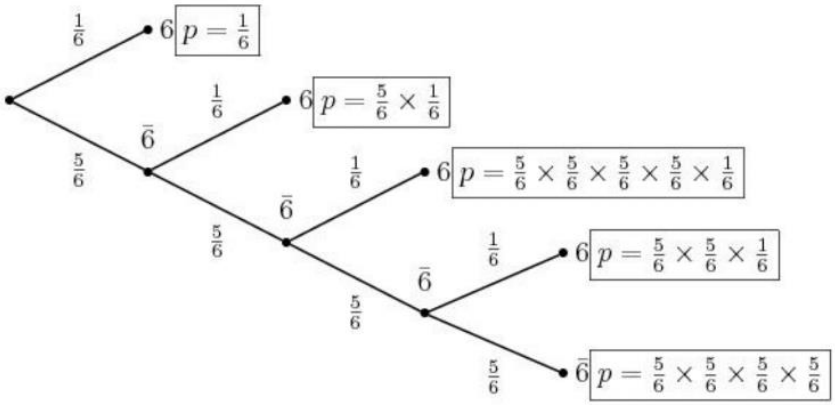


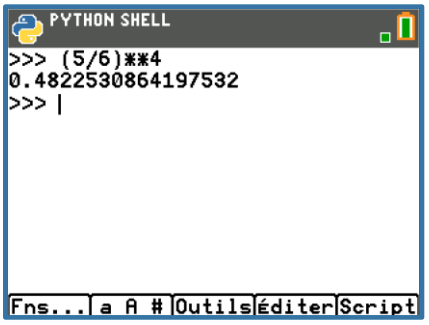
La fonction **prop**(n,N) est à manier avec précaution ! dans la mesure où **prop**(1000,100) demande 100 répétitions de tests sur des échantillons de taille 1000 ... le calcul prendra une vingtaine de secondes.

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



Pour 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 de l'arbre ci-dessous :





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