

# Programmer une expérience aléatoire Niveau 2

## Leçon



Le logiciel Scratch permet de réaliser des simulations d'expériences aléatoires grâce à l'opérateur « nombre aléatoire entre ... et ... ».

Ceci va se révéler très utile pour traiter des problèmes de probabilité, par exemple en répétant un grand nombre de fois les expériences !

### 1. Approcher une probabilité à l'aide de la fréquence.

On considère une expérience aléatoire et un évènement dont on ne connaît pas la probabilité. Comme tu l'as vu en classe, il est possible d'approcher cette **probabilité** par la **fréquence** d'apparition de cet évènement. Plus le nombre de répétitions sera **grand** et plus la fréquence observée sera **proche** de la probabilité réelle !

Exemple : On considère un dé équilibré à 20 faces et l'on s'intéresse à l'évènement A « tirer un nombre supérieur ou égal à 15 ».



Grâce au programme suivant on va simuler la répétition d'un grand nombre de lancers puis calculer la fréquence de réalisation de l'évènement A.

The Scratch script is as follows:

- when green flag clicked
- ask "Combien de répétitions ?" and wait
- set "Nombre de lancers" to response
- set "Nombre de succès" to 0
- repeat "Nombre de lancers" times
  - set "Valeur du dé" to random number between 1 and 20
  - if "Valeur du dé" > 14 then
    - add 1 to "Nombre de succès"
- say "regrouper La fréquence de succès est de : " et "Nombre de succès / Nombre de lancers"

Annotations:

- On introduit les variables :
  - Nombre de lancers (en le demandant à utilisateur)
  - Nombre de succès
- On répète le nombre de fois demandé le tirage
- On tire au hasard un nombre en 1 et 20  
Si ce nombre est supérieur ou égal à 15 (donc > 14) on le compte comme un succès.
- On calcule et affiche la fréquence d'apparition des succès, par rapport au nombre total de lancers !

Il y a ici 6 issues réalisant A (15, 16, 17, 18, 19, 20) sur les 20 possibles. Puisque toutes sont équiprobables on a :  $P(A) = \frac{6}{20} = 0,3$ .

On obtient avec Scratch :

La **fréquence** semble bien se **stabiliser** autour de la **probabilité** réelle lorsque le nombre de lancers augmente !

La fréquence de succès est de : 0.37

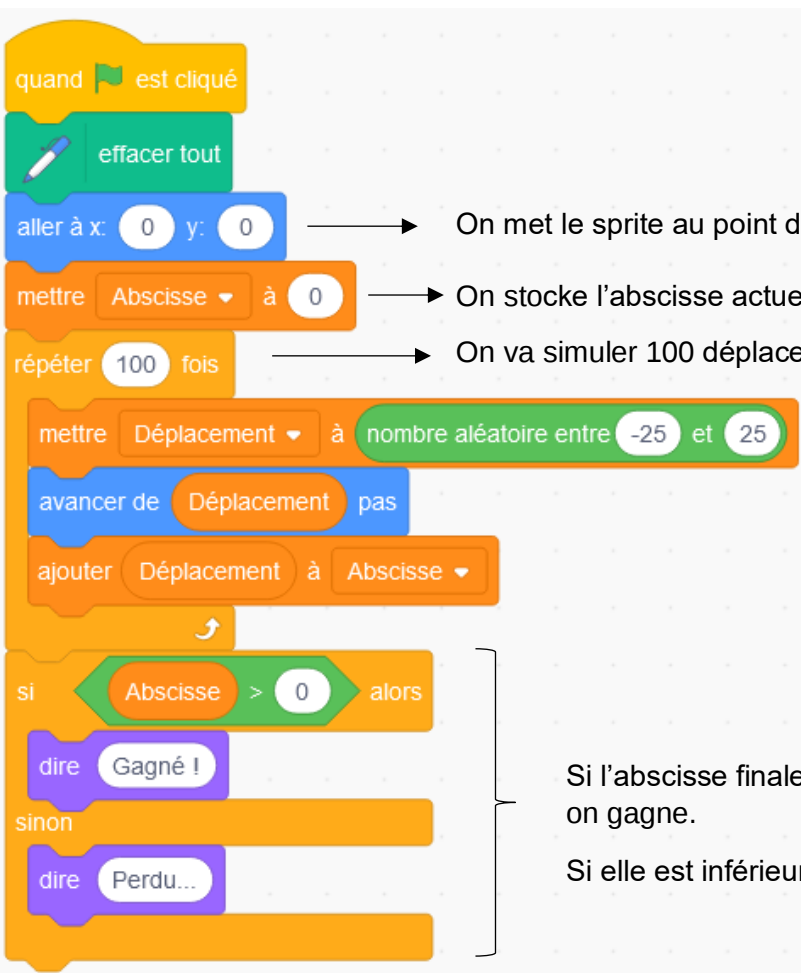
Nombre de lancers = 100

La fréquence de succès est de : 0.326

= 1 000

## 2. Effectuer des déplacements aléatoires

Il est possible d'effectuer des expériences dans lesquelles on va déplacer le sprite aléatoirement ! Par exemple, voici un programme qui va faire avancer 100 fois le sprite horizontalement d'un nombre de pas compris entre -25 et 25 (donc potentiellement reculer !). On considèrera que l'on gagne si le Sprite a avancé après ses 100 déplacements et perdu s'il a reculé !



The Scratch script is as follows:

- when green flag clicked
- clear all
- go to x: 0 y: 0
- set x coordinate to 0
- repeat 100 times
  - set displacement to random number between -25 and 25
  - move displacement steps
  - add displacement to x coordinate
- if x coordinate > 0 then
  - say "Gagné !" for 2 seconds
- else
  - say "Perdu..." for 2 seconds

Annotations:

- On met le sprite au point de coordonnées (0 ; 0)
- On stocke l'abscisse actuelle du sprite dans cette variable.
- On va simuler 100 déplacements
- On tire au hasard le nombre de pas de déplacements et on fait avancer le sprite. On conserve dans la variable « abscisse » l'abscisse du sprite après son déplacement.
- Si l'abscisse finale est supérieure à 0 (le sprite a avancé) on gagne.
- Si elle est inférieure à 0 il a reculé et on perd !

Voici un exemple de simulation. On voit que le dernier déplacement a été de 5 et que l'abscisse finale est de 15 > 0 : on a donc gagné !



## Travaillez cette notion avec un parcours pédagogique progressif clé en main

Cette ressource de type : **Cours gratuit**, pour le niveau : **3ème**, dans la catégorie : **Scratch**, ne constitue qu' **une étape** d'un **parcours pédagogique complet prêt à l'emploi** pour apprendre, s'entraîner et maîtriser cette notion **étape par étape**.

### Étape 1. Cours gratuit : Comprendre la règle

Découvrir la notion grâce à des explications simples et des exemples concrets.

- [Cours Programmer une expérience aléatoire : Secondaire 3 - Scratch](#) (pdf)
- [Cours Programmer une expérience aléatoire : Secondaire 3 - Scratch](#) (word)
- [Télécharger toutes les fiches de cette étape](#) (zip)

### Étape 2. Exercices : S'entraîner progressivement

Des exercices progressifs pour avancer étape par étape

- [Exercices Programmer une expérience aléatoire : Secondaire 3 - Scratch](#) (pdf)
- [Exercices Programmer une expérience aléatoire : Secondaire 3 - Scratch](#) (word)
- [Correction Exercices Programmer une expérience aléatoire : Secondaire 3 - Scratch](#) (pdf)
- [Correction Exercices Programmer une expérience aléatoire : Secondaire 3 - Scratch](#) (word)
- [Télécharger toutes les fiches de cette étape](#) (zip)

### Étape 3. Séquence / Fiche de prep : Ressources enseignants clé en main

Accéder aux fiches de préparation, séquences et séances complètes pour enseigner la notion facilement.

- [Programmer une expérience aléatoire - Scratch - Séquence complète : Secondaire 3](#)
- [Télécharger toutes les fiches de cette étape](#) (zip)