

L'ATTRACTION FOLLE

TI-Nspire™ - TI-Nspire™ CAS

Mots-clés : aléatoire, simulation, programme, matrice.

1. Objectifs

- Approcher un problème de marche aléatoire.
- Utiliser différents registres pour comprendre le phénomène (simulation, matrices, programmation).

2. Énoncé

A la fête foraine, une nouvelle attraction vient de voir le jour. Une roue pouvant pivoter autour de son axe, est partagée en 6 secteurs égaux. Un seul secteur, le rouge, donne droit à un lot important. Les autres ne correspondent qu'à des lots de consolation. Pour gagner le gros lot, il suffit qu'à l'issue de la partie, le secteur rouge soit positionné en face de la flèche, comme ci-contre. Chaque introduction d'une pièce de 1 € fait tourner la roue, de manière aléatoire, d'un secteur vers la droite (D) ou vers la gauche (G).

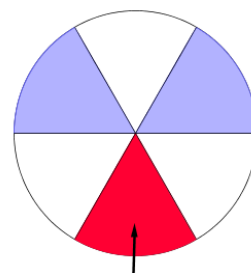


figure 1

3. Conduite de l'activité

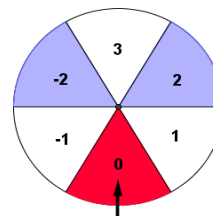
1) Première approche – Un cas particulier à 4 €

On suppose qu'avant que le joueur ne paye le premier euro, la roue est positionnée comme en figure 1.

a) Quel sera le secteur positionné en face de la flèche si la séquence de déplacements est **GDDD** ?

Il peut être plus simple de numérotiser les cases de la roue comme ci-contre et de convenir alors qu'une rotation vers la droite correspond à la valeur +1 et qu'une rotation vers la gauche correspond lui à la valeur -1. Le déplacement donné en exemple s'écrit alors -1,+1,+1,+1 ce qui, si on additionne les 4 nombres, donne +2, numéro de la case finale.

Dans cette façon de procéder, le résultat -3 sera assimilé au résultat +3.



b) Simuler, à l'aide de la calculatrice, une série de 4 rotations successives.

Pour cela, générer, dans la colonne **A** du tableur, une suite de 4 nombres valant chacun +1 ou -1.

Ajouter les éléments de la colonne **A** dans la colonne **B**.

Relancer plusieurs fois la séquence d'instructions (**ctrl** **R**).

Que remarque-t-on sur les résultats ? Expliquer.

	A	B	C
1	$=2*\text{randint}(0,1,4)-1$	1	2
2	-1	-1	
3	1	1	
4	1	1	
5			
6			

B1 =sum(a[1:4])

c) Est-il raisonnable de penser que le joueur puisse gagner le lot principal ?

d) Justifier que dépenser 5 € plutôt que 4, ne constitue pas une bonne stratégie.

e) En quoi le fait de jouer 6 € change-t-il les possibilités de gagner ?

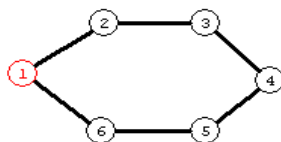
Comment détecter que la somme correspond au secteur rouge (utiliser les congruences) ?

f) Écrire un programme **jeu** qui simule une partie à 6 € et affiche « gagné » ou « perdu », selon les cas.

Remarque : On pourra noter s la variable, initialement à 0, qui donne le numéro du secteur atteint après chaque mouvement.

2) Deuxième approche – Via les matrices et pour toute somme engagée

On considère le graphe à six sommets ci-dessous.



Les sommets sont les six secteurs. Deux sommets sont liés dès lors que les secteurs sont voisins sur la roue (il est possible de passer de l'un à l'autre en un seul mouvement).

On suppose à nouveau que, initialement, le secteur rouge (ici le sommet 1) est en position « gagné ».

a) Donner la matrice de transition a de ce graphe.

b) Rappeler comment, à l'aide des puissances de a , on peut déterminer la probabilité pour qu'un joueur gagne en ayant dépensé exactement n €

c) En déduire la probabilité de gagner en ayant dépensé exactement 4 € et comparer le résultat à votre réponse formulée en 1c.

d) Le joueur est-il assuré de gagner, quitte à dépenser une somme importante ?

e) Désormais, le chariot est positionné de manière aléatoire à l'arrivée du joueur.

Comment, via la matrice a , peut-on obtenir la probabilité de gagner après exactement n rotations de la roue ?

f) Que faut-il modifier dans le cas où la roue à 1 chance sur 5 de rester immobile, 2 chances sur 5 de tourner d'un secteur à droite et 2 chances sur 5 de tourner d'un secteur à gauche ?

3) Troisième approche – Un algorithme qui détecte la victoire.

Un joueur espère arriver à ses fins en engageant 20 euros au maximum. Il peut donc activer le chariot jusqu'à 20 fois mais peut décider de s'arrêter à l'issue de n'importe lequel des 20 mouvements.

Initialement, le chariot est positionné sur la case 0.

a) **A l'aide du tableur**

Générer une liste de 20 entiers égaux à +1 ou -1 dans la colonne **A**.

La TI-Nspire dispose d'une instruction **cumulativesum**, permettant d'obtenir dans la colonne **B**, de proche en proche, les sommes des entiers de la colonne **A**.

Comment le joueur détecte-t-il les passages du chariot en position 0 ?

Renouveler la simulation en appuyant sur **ctrl** **R**.

b) Avec un programme

Écrire un programme *gain* qui donne le nombre d'euros nécessaires pour gagner.

Exemple d'algorithme possible

Initialiser p à 1 ou -1 (p : position après le premier euro)

Initialiser n à 1 (n : nombre de rotations)

Tant que le chariot n'est pas revenu en position initiale et que l'on a dépensé 20 € ou moins faire

Ajouter 1 ou -1 à p

Augmenter n de 1

Fin tant que

Si le secteur rouge a été atteint

Afficher Gagné en n euros

Sinon

Afficher Perdu

Fin Si

Remarque : Le fait d'initialiser p à ± 1 , et donc n à 1, revient à commencer la boucle après le premier mouvement de la roue. Il reste donc au maximum 19 mouvements à faire.

c) Écrire un programme *cumul*, d'argument n , permettant de sommer les résultats lors de n exécutions du programme *gain* et qui affiche la moyenne obtenue.

Attention ! Le programme *gain* affiche du texte que l'on ne souhaite pas voir à chaque répétition. On utilisera donc plutôt un programme *gain2*, dans lequel le texte aura été supprimé.