

La marelle de Bresenham

La marelle de Bresenham se joue à trois (Alice, Benjamin et Céline) sur un carrelage rectangulaire à grandes dalles orienté Ouest-Est en longueur et Sud-Nord en largeur. La dalle de départ est la dalle Sud-Ouest de la pièce (en bas à gauche, si on représente le carrelage par un rectangle). Alice choisit un entier naturel a et Benjamin un entier naturel b .

Au début de chaque partie Céline est sur la dalle de départ et tient un compteur (ou score) s qui est initialement à zéro. Lorsque le jeu démarre Céline se déplace suivant la règle suivante :

- Si le compteur est strictement négatif, elle avance d'un carreau vers l'Est et le compteur augmente de a .
- Si le score est positif ou nul, elle avance d'un carreau vers le Nord et le compteur diminue de b . Le jeu s'arrête lorsqu'elle arrive sur un bord du carrelage.

1°) Le cas 10×10

On s'intéresse dans cette question à un carrelage de 10 par 10 que l'on modélise par un quadrillage. La position de Céline est repérée par ses coordonnées x et y dans ce quadrillage. Au début du jeu, on a donc $(x; y) = (0; 0)$.

On décide de simuler la marelle à l'aide d'un algorithme. On propose les algorithmes 1 et 2 suivants.

Algorithme 1

Initialisations :

x prend la valeur 0
 y prend la valeur 0
 s prend la valeur 0

Traitement et sorties :

a prend la valeur d'un entier aléatoire compris entre 0 et 11 au sens large
 b prend la valeur d'un entier aléatoire compris entre 0 et 11 au sens large

Tantque ($x \leq 10$ et $y \leq 10$) **Faire**

Afficher le point de coordonnées $(x; y)$

Si $s < 0$

Alors x prend la valeur $x + 1$

s prend la valeur $s + a$

Sinon

y prend la valeur $y + 1$

s prend la valeur $s - b$

FinSi

FinTantque

Algorithme 2

Initialisations :

x prend la valeur 0
 y prend la valeur 0
 s prend la valeur 0

Traitement et sorties :

Tantque ($x \leq 10$ et $y \leq 10$) **Faire**

a prend la valeur d'un entier aléatoire compris entre 0 et 11 au sens large
 b prend la valeur d'un entier aléatoire compris entre 0 et 11 au sens large
Afficher le point de coordonnées (x ; y)

Si $s < 0$

Alors x prend la valeur $x+1$
 s prend la valeur $s+a$

Sinon

y prend la valeur $y+1$
 s prend la valeur $s-b$

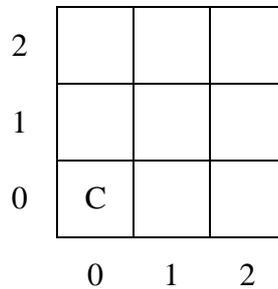
FinSi

FinTantque

Déterminer l'algorithme modélisant la marelle de Bresenham sur un quadrillage de 10 par 10.
Programmer cet algorithme sur calculatrice ou avec le logiciel Algobox.

2°) Le cas 3×3

On s'intéresse dans cette question à la petite marelle de Bresenham représentée ci-dessous.



a) En la reproduisant sur une feuille, tester le déplacement de Céline en comptant le nombre de déplacements dans les cas suivants :

- en choisissant $a = b$ (faire plusieurs essais) ;
- en choisissant $a = 1$ et $b = 10$.

b) Démontrer que le premier déplacement de Céline ne dépend pas du choix de a et de b .

Comment cette remarque permet-elle de modifier l'algorithme 1 vu précédemment pour diminuer le nombre de boucles faites ?

3°) Pertinence de l'algorithme

On s'intéresse cette fois à la finitude de l'algorithme. La boucle utilisée ne s'arrêtant que si une condition est remplie, il faut prouver que cette condition se réalisera en un nombre fini d'étapes.

Pour cela, on se place dans la condition initiale de la marelle de 10 sur 10.

a) On suppose que Céline a atteint la frontière Est après 17 pas.

Combien a-t-elle fait de pas vers l'Est ? Vers le Nord ?

b) Même question avec 25 pas.

c) Que peut-on en conclure ?

d) Si cette fois la marelle est de n par p (avec n et p deux entiers tels que $2 \leq n \leq p$), quel est le nombre maximal de boucles faites par l'algorithme ?

Corrigé

1°) L'algorithme 1 ne convient pas car les nombres aléatoires ne sont pas intégrés à la boucle ; ils ne seront définis qu'une seule fois.

Il n'y aura pas d'évolution.

Deux propositions de programmes sur calculatrice TI (à gauche, modèle TI-83 Premium-CE)

```
: 0 → X
: 0 → Y
: 0 → S
: nbrAléatEnt(0,11) → A
: nbrAléatEnt(0,11) → B
: While X ≤ 10 et Y ≤ 10
: Pt-Aff(X,Y)
: If S < 0
: Then
: X + 1 → X
: S + A → S
: Else
: Y + 1 → Y
: S - B → S
: End
: End
```

```
: ClrDraw
: AxesOff
: FnOff
: 0 → Xmin
: 10 → Xmax
: 0 → Ymin
: 10 → Ymax
: 0 → X
: 0 → Y
: 0 → S
: While X ≤ 10 and Y ≤ 10
: randInt(0,11) → A
: randInt(0,11) → B
: Pt-On(X,Y)
: If S < 0
: Then
: X + 1 → X
: S + A → S
: Else
: Y + 1 → Y
: S - B → S
: End
: End
```