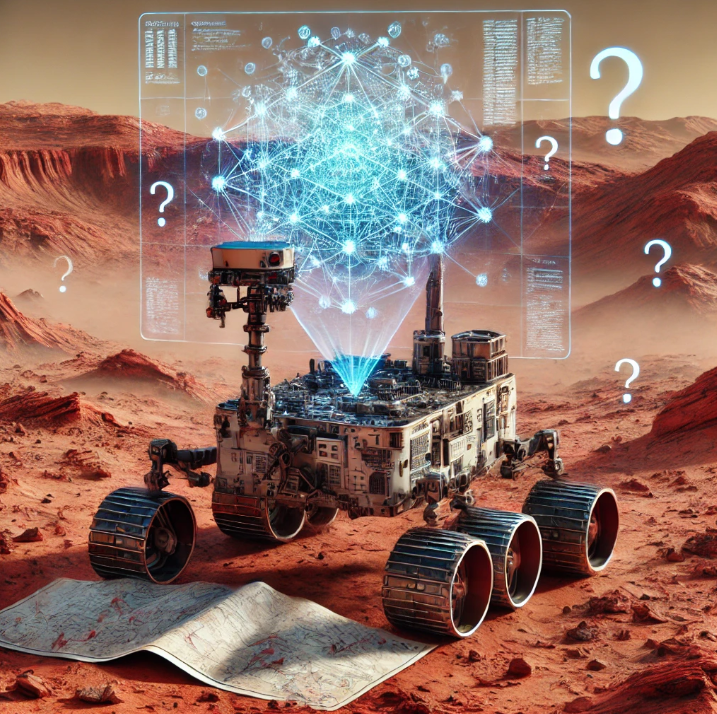
# Rover Run !



**Note** : les illustrations de ce document (page de garde, pictogrammes et exemples) ont été réalisées à l’aide de l’IA générative DALL.3 Ultra

# Contexte de la mission

2028 : Une tempête solaire très intense vient de se produire… Le rover MARC – MArs Rover Cartograph, qui s’acquittait fort bien de sa mission, a été victime d’un dysfonctionnement. Ses systèmes de programmation, de guidage et de mouvements ont été fortement altérés…

L’ESTF, Efrei Space Task Force, a fait appel à des experts en programmation pour concevoir un nouveau logiciel de déplacement pour MARC.

Votre mission est de réaliser ce logiciel de déplacement pour ramener MARC jusqu’à une station de base où il pourra être remis en état.

# Le rover MARC

MARC est un rover un peu rustique, il embarque un système Linux, et tous ses programmes sont écrits en langage C.

Il se repère par sa position, et la direction à laquelle il fait face, parmi Nord, Sud, Est et Ouest (même sur Mars). Fort heureusement, son radar et son gyroscope fonctionnent encore et lui permettent de se situer.

## Système de déplacement

MARC peut effectuer les mouvements suivants (ce sont des mouvements unitaires)

* avancer de 10m
* avancer de 20m
* avancer de 30m
* reculer de 10m (en marche arrière)
* tourner d’un quart de tour à gauche
* tourner d’un quart de tour à droite
* faire demi-tour.

Les avaries de ses systèmes font que son déplacement se déroule maintenant par phases, et il ne peut donc pas directement aller d’un point A à un point B.

Pour chaque phase, MARC disposera de 9 mouvements possibles (tirés au hasard via la table 1 – probabilité de mouvement disponible), et devra en choisir **exactement** 5 pour réaliser cette phase. Il va donc falloir programmer le « meilleur choix possible » de déplacement.

|  |  |
| --- | --- |
| **Type de mouvement unitaire** | **Probabilité de tirage** |
| Avancer de 10 m | 22% |
| Avancer de 20 m | 15% |
| Avancer de 30 m | 7 % |
| Reculer de 10 m en marche arrière | 7 % |
| Tourner d’un quart de tour à gauche | 21 % |
| Tourner d’un quart de tour à droite | 21 % |
| Faire demi-tour | 7 % |
| **TOTAL** | 100 % |

**Le programme de tirage au hasard est buggé – chaque fois qu’un mouvement est choisi, sa probabilité de tirage diminue de 1%**

Table 1 – Probabilité de tirage d’un mouvement

La procédure de tirage des 9 mouvements est donc la suivante :

Répéter 9 fois

Tirer un mouvement au hasard

Réduire la probabilité de tirage de ce mouvement de 1%

Les probabilités de tirage retrouvent leurs valeurs initiales au début de chaque phase.

# Voyager sur Mars

Puisque MARC ne se déplace que sur le sol martien, la carte de Mars est un espace à deux dimensions : cette carte est constituée de zones de 10 m x 10 m, que l’on peut représenter par une ‘case’ carrée. On repèrera une case par des coordonnées x et y sur la carte.

## Le paysage martien

La composition et le relief du sol martien influencent les déplacements du rover à sa surface. Les types de sol identifiés sont :

**Plaine** : aucune influence sur le mouvement

**Erg martien** : sol meuble et poussiéreux. **Si MARC commence son mouvement depuis une case Erg**, son prochain mouvement est diminué de 1 : Avancer de 10m et Reculer en marche arrière ne font rien, avancer de 20m n’avance que de 10m, avancer de 30m n’avance que de 20m. On ne peut pas tourner par quart de tour, le demi-tour fait tourner, au choix, à gauche ou à droite d’un quart de tour.

**Reg martien** : sol accidenté. **Si MARC termine un mouvement sur une case Reg**, il a été secoué et son logiciel fonctionne moins bien. Il n’aura droit qu’à 4 mouvements en tout pour la phase suivante.

**Crevasse** : zone très peu recommandée pour les rover : **si MARC passe sur une case crevasse**, il y tombe et termine sa vie de rover, sans respawn.

**Pente**  – la case descend dans une direction donnée : si MARC termine son mouvement sur une case pente, il est ensuite déplacé d’une case dans le sens de la pente descendante.

On peut combiner **pente+plaine**, **pente+erg** (c’est une dune en fait), **pente+reg**

# Choix du meilleur parcours

Sur la carte de Mars, la station de base sera située à des coordonnées prédéfinies et , le point de départ de MARC sera situé à des coordonnées arbitraires et .

Votre mission est donc, au fur et à mesure des phases, de faire en sorte que MARC arrive à rejoindre la station de base, puis dans un second temps, de minimiser le nombre de déplacements utilisés.

Le choix d’un meilleur parcours est effectué par la minimisation d’une fonction de coût.

## Fonction de coût de parcours – cette partie vous est fournie

La fonction de coût sera précalculée sur la carte.

A chaque case sera associé une valeur (un coût), d’autant plus faible que cette case sera ‘intéressante’.

Par exemple, et pour convention : la case où est située la station de base vaudra 0 (le plus petit coût possible donc très intéressant), et les cases crevasses seront associées à un coût très élevé (par exemple : 10000).

Ainsi, le programme de guidage de MARC cherchera à atteindre les cases de valeurs minimales tout au long de son parcours.

### Principe de calcul de la fonction de coût – étape 1

1. Initialiser le coût de toutes les case à une grande valeur (par exemple 65 535).
2. Initialiser le coût de la case de la station de base à 0.
3. En partant de la station de base (valeur 0), mettre à jour ses 4 voisins (car MARC ne se déplace pas en diagonale), en ajoutant un coût, relatif à la nature de cette case (plaine, erg, reg).

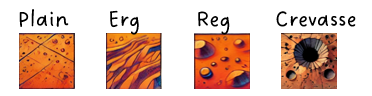
Le **coût final d’une case** est alors :

**coût associé à la nature de la case + min(coûts de ses 4 voisines).**

Par exemple, une plaine est un terrain a priori plus accessible et manœuvrable qu’un reg, lui-même plus ‘intéressant’ qu’un erg. Voici un exemple de coûts (vous pouvez modifier ces valeurs)

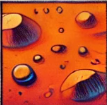
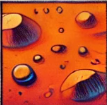
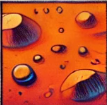
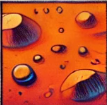
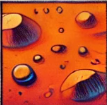
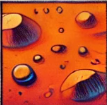
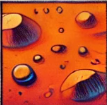
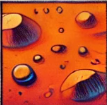
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type de case | plaine | reg | erg | crevasse |
| Coût associé | 1 | 2 | 4 | 10 000 |

Illustration :

Les types de terrain étant représentés par les dessins suivants

On s’intéresse à la carte suivante, où sont également visualisés : la station de base, et la position initiale de MARC (sur une plaine). A droite, les coûts initiaux :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 65535 | 65535 | 65535 | 65535 | 65535 | 65535 |
| 65535 | 65535 | 0 | 65535 | 65535 | 65535 |
| 65535 | 65535 | 65535 | 65535 | 65535 | 65535 |
| 65535 | 65535 | 65535 | 65535 | 65535 | 65535 |
| 65535 | 65535 | 65535 | 65535 | 65535 | 65535 |
| 65535 | 65535 | 65535 | 65535 | 65535 | 65535 |
| 65535 | 65535 | 65535 | 65535 | 65535 | 65535 |



Etape 1 : voisines de la base Etape 2 : voisines des voisines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 65535 | 5 | 4 | 2 | 65535 | 65535 |
| 5 | 1 | 0 | 1 | 3 | 65535 |
| 65535 | 2 | 1 | 3 | 65535 | 65535 |
| 65535 | 65535 | 10001 | 65535 | 65535 | 65535 |
| 65535 | 65535 | 65535 | 65535 | 65535 | 65535 |
| 65535 | 65535 | 65535 | 65535 | 65535 | 65535 |
| 65535 | 65535 | 65535 | 65535 | 65535 | 65535 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 65535 | 65535 | 4 | 65535 | 65535 | 65535 |
| 65535 | 1 | 0 | 1 | 65535 | 65535 |
| 65535 | 65535 | 1 | 65535 | 65535 | 65535 |
| 65535 | 65535 | 65535 | 65535 | 65535 | 65535 |
| 65535 | 65535 | 65535 | 65535 | 65535 | 65535 |
| 65535 | 65535 | 65535 | 65535 | 65535 | 65535 |
| 65535 | 65535 | 65535 | 65535 | 65535 | 65535 |

## Etat final de l’étape 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 7 | 5 | 4 | 2 | 10002 | 7 |
| 5 | 1 | 0 | 1 | 3 | 5 |
| 3 | 2 | 1 | 3 | 4 | 5 |
| 6 | 3 | 10001 | 5 | 6 | 6 |
| 10 | 7 | 10003 | 7 | 7 | 7 |
| 12 | 8 | 10004 | 8 | 8 | 8 |
| 16 | 12 | 10007 | 10008 | 9 | 9 |

### Principe de calcul de la fonction de coût – étape 2

Pour les cases de coût supérieur à 10000 **et qui ne sont pas des crevasses** (nombres en rouge dans le tableau), refaire le même calcul en commençant par le coût le plus bas : on recalcule le coût de la case en fonction de sa nature + coût minimum de ses 4 voisins.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 6 | 5 | 4 | 2 | 10002 | 7 |
| 5 | 1 | 0 | 1 | 3 | 5 |
| 3 | 2 | 1 | 3 | 4 | 5 |
| 6 | 3 | 10001 | 5 | 6 | 6 |
| 10 | 7 | 8 | 7 | 7 | 7 |
| 12 | 8 | 9 | 8 | 8 | 8 |
| 16 | 12 | 13 | 10008 | 9 | 9 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 6 | 5 | 4 | 2 | 10002 | 7 |
| 5 | 1 | 0 | 1 | 3 | 5 |
| 3 | 2 | 1 | 3 | 4 | 5 |
| 6 | 3 | 10001 | 5 | 6 | 6 |
| 10 | 7 | 10003 | 7 | 7 | 7 |
| 12 | 8 | 10004 | 8 | 8 | 8 |
| 16 | 12 | 10007 | 10008 | 9 | 9 |

## Sélection des déplacements de phase – cette partie n’est pas fournie

Une phase est donc constituée de 5 mouvements (4 dans le cas d’une case **Reg**) choisis par MARC.

Pour faire le ‘meilleur’ choix possible pour une phase, on utilisera un **arbre N-aire**, dont chaque niveau représente un des déplacements de la phase. Le niveau 0 (la racine) n’est pas un déplacement, mais sert à débuter le parcours de l’arbre. Le chemin ‘optimal’ par rapport à une phase donnée sera la feuille de valeur minimale de cet arbre.

### Structure d’arbre N-aire

Pour le niveau 1 (choix du premier mouvement), on a 9 possibilités.

Pour le niveau 2, (choix du deuxième mouvement), on a 8 possibilités, pour chaque premier choix fait , soit 8\*9 nœuds)

Pour le niveau 3, on a 7 possibilités pour chaque nœud du niveau précédant, soit 7\*8\* nœuds…

En tout, on a donc : 1 + 9 +9\*8 +9\*8\*7+9\*8\*7\*6+9\*8\*7\*6\*5 nœuds = 18730 nœuds (dont 9\*8\*7\*6\*5 = 15120 feuilles).

### Valeurs stockées dans les nœuds

Les valeurs stockées dans cet arbre seront : pour la racine, la valeur de la case de départ de la phase de mouvement de MARC. Pour chaque descendant d’un nœud, la valeur est celle de la case atteinte après application du déplacement associé au mouvement choisi.

#### Méthode numéro 1- obligatoire

Construire l’arbre entier correspondant. Trouver, parmi les feuilles, la valeur minimale obtenue. Trouver la (ou une) feuille qui stocke cette valeur. Retrouver (par un moyen à définir par vous-même) la séquence de mouvements associée à cette feuille.

#### Méthode numéro 2 - facultative

Parcourir d’abord le chemin ‘le plus prometteur’ : plutôt que de construire tout l’arbre, on effectue les choix à partir du nœud de plus petite valeur trouvé à un instant donné. On finit par trouver une feuille de valeur minimale. Retrouver (par un moyen à définir par vous-même) la séquence de mouvements associée à cette feuille.

### Cas exceptionnels – arrêt avant fin de phase

Si, à la fin d’un mouvement quelconque, (donc pas forcément en fin de phase), MARC arrive à la station de base, on peut stopper la phase (si vous trouvez un nœud de valeur 0, c’est gagné !).

Si MARC ‘sort’ de la carte, on perd le contact avec lui, et il termine sa vie de rover sur Mars.

### Mise à jour de la position et de l’orientation de MARC suite au choix des déplacements – la mise à jour des coordonnées de MARC est fournie

Appliquer la séquence de mouvements retenue pour mettre à jour la position et l’orientation de MARC.

## Illustrations pour la construction de l’arbre

Pour simplifier, l’exemple sera pris pour un choix de 3 mouvements parmi 5. La situation de départ est celle de la carte suivante (déjà prise en exemple). MARC est positionné en bas à droite de la carte et fait face au ‘NORD’, vers le haut de la carte. Sa case de départ a une valeur de 9.

Mouvements disponibles tires au hasard

Avancer de 10 m

Quart de tour à gauche

Quart de tour à droite

Avancer de 20 m

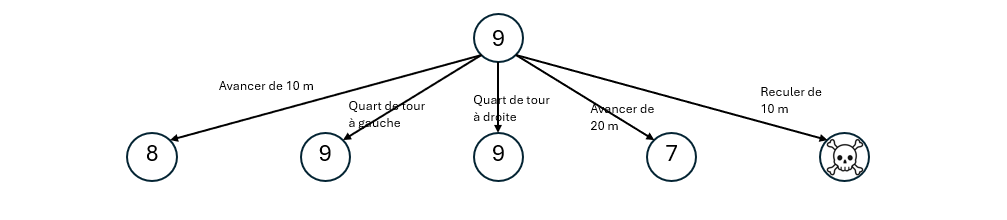
Reculer de 10 m

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 6 | 5 | 4 | 2 | 10002 | 7 |
| 5 | 1 | 0 | 1 | 3 | 5 |
| 3 | 2 | 1 | 3 | 4 | 5 |
| 6 | 3 | 10001 | 5 | 6 | 6 |
| 10 | 7 | 8 | 7 | 7 | 7 |
| 12 | 8 | 9 | 8 | 8 | 8 |
| 16 | 12 | 13 | 10008 | (M ↑) 9 | 9 |

Etat initial de l’arbre : la racine contient la valeur de départ

9

Premier choix de mouvement : 5 possibilités. Le fils le plus à gauche correspond au score de la case atteinte par le déplacement ‘Avancer de 10 m’, c’est la case valant 8



Note : Une image contenant croquis, dessin, illustration, dessin humoristique

Description générée automatiquement vaudra, comme pour une crevasse, 10 000. Donc, dans l’arbre, on aura : si la valeur du nœud est plus grande que 9999, on ne descend pas plus bas pour ce nœud.

Deuxième choix : illustration à partir du nœud ‘8’ : le déplacement ‘Avancer de 10 m’ a été utilisé, on essaie avec les 4 autres restants

Une image contenant diagramme, ligne, cercle

Description générée automatiquement

Après 5 choix, vous aurez construit toutes les séquences possibles de déplacement. la (les) meilleures séquences sera (seront) celle(s) qui amènent vers la case de coût le plus bas, le résultat final étant dans une feuille.

Il faut donc sélectionner une feuille de valeur minimale.

Ensuite, à partir de cette feuille, il faut déterminer le chemin suivi depuis la racine de l’arbre pour y arriver, ce qui vous donnera la suite des mouvements à effectuer.

# Pour réussir votre mission

Procédez par étapes successives

## Etape 1- obligatoire

* Tester le code fournir pour les cartes et les déplacements individuels de MARC ;
* Définir les structures de données pour l’arbre de sélection des déplacements ;
* Commencer simple : créer un arbre avec 3 choix parmi 4 ou 5 valeurs possibles, pour bien mettre au point les algorithmes de **construction** de l’arbre, de **recherche d’une feuille de valeur minimale**, et de **chemin depuis la racine vers cette feuille**.

## Etape 2 - obligatoire

Passage à l’échelle : augmentez progressivement le nombre de choix à faire, et le nombre de valeurs possibles, jusqu’à arriver au choix de 5 mouvements parmi 9 possibilités.

Complexité : indiquer les temps d’exécution de votre programme pour :

* La phase de construction de l’arbre ;
* La phase de recherche d’une feuille de valeur minimale parmi toutes les feuilles de l’arbre ;
* Le calcul du chemin de la racine vers cette feuille ;
* Un exemple complet de guidage de MARC depuis sa position d’origine vers la station de base.

Inventez vos propres cartes géographiques de Mars pour voir comment MARC s’y débrouille, grâce à votre programme

## Etape 3 - facultative

Etude de complexité : maintenant, on cherche à trouver le meilleur compromis temps / performance de parcours par MARC. Continuez à augmenter le nombre de choix de mouvements pour une phase, ainsi que le nombre de possibilités de choix.

Indiquez quelle est le temps de calcul des algorithmes au fur et à mesure que l’arbre grandit.

Indiquez si ces augmentations donnent de meilleurs résultats pour le parcours effectué par MARC sur une carte.

Idée d’extensions : implémenter les terrains en pente, ajouter de nouveaux types de terrains, améliorer l’interface graphique.