

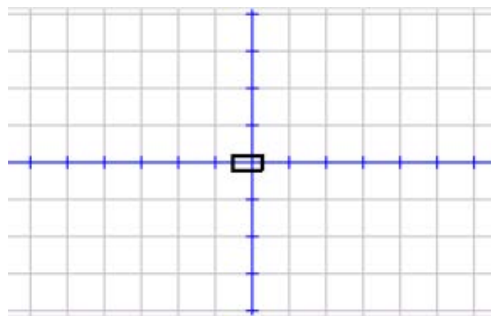


Dans cette première leçon de l'unité 6, vous allez apprendre à connaître le système de coordonnées utilisé par le TI-Innovator Rover afin de le mettre en mouvement.

Objectifs :

- Comprendre le système de coordonnées du Rover, la position initiale et le cap.
- Mettre le Rover en mouvement vers un point de coordonnées fixées.
- Utiliser les mathématiques pour calculer une distance.

Le Rover dispose d'un système de coordonnées intégré exactement comme un repère. Lorsque vous envoyez l'instruction **Send "CONNECT RV"**, la position du Rover sur une grille virtuelle est (0,0) et son cap est de 0 degrés (vers l'axe des abscisses positives).



FORWARD met le robot en mouvement vers la droite.

LEFT tourne le Rover de 90 degrés dans le sens anti horaire.

Notre programme demandera au Rover d'atteindre un point de coordonnées données sur sa grille. Nous utiliserons l'instruction **Request** pour entrer les valeurs de x et y et faire en sorte que le Rover atteigne le point de coordonnées (x, y) et retourne ensuite à son origine.

1. Commencer votre nouveau programme avec les instructions habituelles.
2. Insérer l'instruction **Request** pour x et y.

Souvenez-vous que l'instruction **Request** affichera un message entre guillemets (le « prompt »).

3. L'instruction **Text** vous laisse le temps de positionner le Rover à l'origine et la face avant orientée dans la bonne direction.

```

1.1 *Classeur RAD
*rover61
Define rover61()=
Prgm
Send "CONNECT RV"
Request "Abscisse x",x
Request "Ordonnée y",y
Text "Appuyer sur Enter pour commencer"
:
:
:
EndPrgm

```

Conseil à l'enseignant : Les étudiants doivent être familiarisés avec le système de coordonnées cartésien. Si tel n'est pas le cas, alors c'est l'occasion parfaite pour reprendre cette notion. La "bonne direction" est celle pour laquelle le robot est placé selon l'axe des x positifs (orienté à l'Est)



10 Minutes de Code

TI-NSPIRE™ CX AVEC LE TI-INNOVATOR™ ROVER

- Ajouter l'instruction **TO XY**. Celle-ci se trouve dans le menu **menu > Hub > Rover (RV) > Drive RV** et apparaîtra dans votre programme non complétée.

Send "RV TO XY "

- Les coordonnées x et y peuvent être ajoutées et seront sauvegardées respectivement dans les variables x et y . Afin que le TI-Innovator™ Hub utilise ces valeurs, vous devez utiliser la fonction **eval()** deux fois.

- Ajouter les instructions **eval(x) eval(y)** à la commande.

Send "RV TO XY eval(x) eval(y)"

- Tester maintenant votre programme. Le Rover doit atteindre le point de coordonnées que vous avez fixé.

Conseil de l'enseignant : Le robot se comporte de la même façon, indépendamment du paramètre de la commande **XYLINE** placé à la fin de l'instruction **RV TO XY**. **XYLINE** se trouve dans le menu **RV Settings...**

Send("RV TO XY eval(X) eval(Y) XYLINE")

- Ajouter une commande **Wait** afin de laisser au Rover le temps d'atteindre votre point, puis de retourner à son point de départ. Utiliser simplement les nombres 0 et 0 séparés par un espace. Ajouter par ailleurs une instruction afin de faire retourner le Rover à sa position d'origine (**TO ANGLE 0**).

- Tester de nouveau votre programme. Cette fois, le Rover devrait atteindre le point fixé, puis retourner à sa position d'origine et se placer selon le cap initial.

UNITE 6 : COMPETENCE 1

NOTES DU PROFESSEUR

```

1.1 *Classeur RAD 5/7
rover61
Define rover61()=
Prgm
Send "CONNECT RV"
Request "Abscisse x",x
Request "Ordonnée y",y
Text "Appuyer sur Enter pour commencer"
Send "RV TO XY "
EndPrgm

```

```

1.1 *Classeur RAD 5/5
rover61
Define rover61()=
Prgm
Send "CONNECT RV"
Request "Abscisse x",x
Request "Ordonnée y",y
Text "Appuyer sur Enter pour commencer"
Send "RV TO XY eval(x) eval(y)"
EndPrgm

```

```

1.1 *Classeur RAD 8/8
rover61
Prgm
Send "CONNECT RV"
Request "Abscisse x",x
Request "Ordonnée y",y
Text "Appuyer sur Enter pour commencer"
Send "RV TO XY eval(x) eval(y)"
Wait 5
Send "RV TO XY 0 0"
Send "RV TO ANGLE 0"
EndPrgm

```