

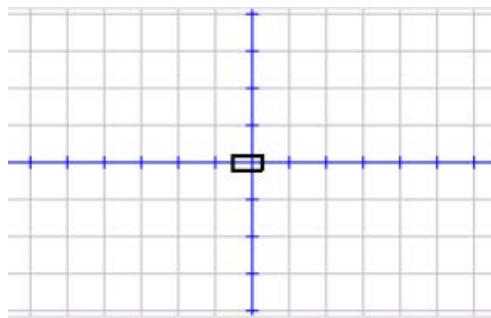


Dans cette première leçon de l'unité 6, vous allez apprendre à connaître le système de coordonnées utilisé par le TI-Innovator Rover afin de le mettre en mouvement.

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 une commande **Prompt** pour entrer les valeurs de X et Y et faire en sorte que le Rover atteigne ce point de coordonnées (X, Y) puis retourne enfin vers l'origine du repère.

1. Commencer votre nouveau programme avec l'instruction EffÉcran (**prgm > E/S > EffÉcran**) qui est une instruction courante.
2. Insérer une instruction **Prompt** pour X et Y.

Souvenez-vous que l'instruction **Prompt** affiche le nom de la variable suivi d'un point d'interrogation. Plus d'une variable peuvent être demandées à l'aide d'une seule instruction **Prompt**.

3. La commande **Pause** vous laisse le temps de positionner le Rover à l'origine et la face avant orientée dans la bonne direction

```
NORMAL FIXE2 AUTO a+b÷ RAD MP
EDIT MENU: [a.1Pho.] [f5]
PROGRAM: ROVER61
:EffÉcran
:Disp "ROVER UNIT6 C1"
:Send("CONNECT RV")
:Prompt X,Y
:Pause "ENTRER POUR COMMEN
CER"
:
:
:
```

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-83 PREMIUM CE AVEC LE TI-INNOVATOR™ ROVER

- Ajouter l'instruction **TO XY** pour mettre le Rover en mouvement :
prgm > Hub > Rover (RV)... > Drive RV... > TO XY

La commande apparaîtra dans votre programme comme une instruction non complétée.

- Les coordonnées *X* et *Y* doivent être ajoutées et sont 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 **eval(X) eval(Y)** sans oublier les guillemets et les parenthèses afin de clore l'instruction.
- Tester votre programme. Le Rover doit atteindre le point de coordonnées que vous avez fixé.

Conseil à 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

```
NORMAL FIXE2 AUTO a+bi RAD MP
EDIT MENU: [a,1pha.] [f5]
```

```
PROGRAM:ROVER61
:Effécran
:Disp "ROVER UNIT6 C1"
:Send("CONNECT RV")
:Prompt X,Y
:Pause "ENTRER POUR COMMEN
CER"
:Send("RV TO XY █
:
:
```

```
NORMAL FIXE2 AUTO a+bi RAD MP
EDIT MENU: [a,1pha.] [f5]
```

```
PROGRAM:ROVER61
:Effécran
:Disp "ROVER UNIT6 C1"
:Send("CONNECT RV")
:Prompt X,Y
:Pause "ENTRER POUR COMMEN
CER"
:Send("RV TO XY eval(X)eva
l(Y)")
:
```

```
NORMAL FIXE2 AUTO a+bi RAD MP
EDIT MENU: [a,1pha.] [f5]
```

```
PROGRAM:ROVER61
:Prompt X,Y
:Pause "ENTRER POUR COMMEN
CER"
:Send("RV TO XY eval(X)eva
l(Y)")
:
:Wait 5
:Send("RV TO XY 0 0")
:Send("RV TO ANGLE 0")
```