

Kapitel 6: Koordinater

I denna lektion kommer du att använda TI-Innovator Rover för att markera två punkter i dess koordinatsystem och bestämma avståndet mellan dessa punkter med avståndsformeln.

Denna aktivitet kräver att en markör används i Rover markörhållare för att lämna ett spår på papperet.

Vi ska skriva ett program som markerar två punkter i planet och sedan beräknar avståndet mellan de två punkterna. Då mäter du avståndet mellan de två punkterna och jämför mätningen med beräkningen.

1. Starta ditt program med **CONNECT RV** och inkludera Rover-kommandot för att ställa in enhetsavstånd till en tum (1 tum \approx 0,0254 meter). **RV.GRID.M/UNIT**-kommandot finns i menyn **Rover (RV) ...> RV Setup**.

Send("SET RV.GRID.M/UNIT .0254")

2. Lägg till fyra **Request**-satser för koordinaterna hos de två punkterna. Kom ihåg att variabler hos TI-Nspire™ CX består av flera tecken så vi kan använda variablerna $x1$, $y1$, $x2$ och $y2$ för koordinaterna.

Request "Första x-koordinaten (x1):",x1

Request "Första y-koordinaten (y1):",y1

Request "Andra x-koordinaten (x2):",x2

Request "Andra y-koordinaten (y2):",y2

Vi använder också **Text**-satsen för att komma igång:

Text "Tryck enter för att starta"

Lärarkommentar: Det förinställda enhetsavståndet är 10 cm. Genom att sätta enhetsavståndet till 1 tum säkerställer att det inte behövs någon skalning för att omvandla från det uppmätta avståndet (i tum) till det beräknade avståndet. "M/UNIT" står för "Meter per Unit", så 0.0254 M/UNIT representerar 1 tum per enhet av rörelse.

3. Sedan får vi Rover att flytta till den första punkten. Därefter kommer satsen **Wait 4** medan Rover rör sig och sedan kommer ljud (**SOUND**) och ljus från lysdioden (**LED COLOR**) att sättas på när Rover har nått sin position. Vi kommer också att använda en **TEXT**-sats här och stänga av lysdioden efter att ha tryckt på **enter**. Kom ihåg att ljudet bara hörs i en sekund.

Övning 2: Avståndsformeln

Syfte:

- Förflyttning till två olika punkter
- Markera punkterna med **SOUND** och **COLOR**, och med en markör
- Beräkna och visa avståndet mellan två punkter
- Mäta avståndet mellan punkterna för att bekräfta beräkningen
- Uppskatta felet i rörelsen kontra beräkningen

```
* rover62 10/10
Prgm
© Avståndsformeln demo
Local x1,y1,x2,y2,d
Send "CONNECT RV"
Send "SET RV.GRID.M/UNIT 0.0254"
Request "Första x-koordinaten (x1) ",x1
Request "Första y-koordinaten (y1) ",y1
Request "Andra x-koordinaten (x2) ",x2
Request "Andra y-koordinaten (y2) ",y2
Text "Tryck enter för att starta. "
```

```
* rover62 13/16
Request "Andra x-koordinaten (x2) ",x2
Request "Andra y-koordinaten (y2) ",y2
Text "Tryck enter för att starta. "
Send "RV TO XY eval(x1) eval(y1) "
Wait 4
Send "SET RV.COLOR 0 200|200"
Send "SET SOUND 256"
Text "Tryck enter för att fortsätta... "
Send "SET RV.COLOR 0 0 0"
```

10 Minutes of Code

TI-Nspire-teknologi med TI-Innovator™ Rover

KAPITEL 6: ÖVNING 2

LÄRARKOMMENTARER

Lärarkommentar: När Rover stannar tenderar markören att lämna ett spår av bläck. Justera då **Wait**-satsen för att styra storleken på bläckfläcken eller kom bara ihåg att det ska finnas ett hörn vid fläcken. Programmet kommer att rita två linjesegment.

4. Därefter får vi Rover att förflytta sig till den andra punkten. Ett annat ljud och en annan färg från lysdioden används för att indikera att Rover har nått punkten. En **Wait**-sats används här för att behålla lysdioden (LED) på innan den stängs av.

Tips: Använd Kopiera och Klistra in istället för att skriva om likartad kod.

```
* rover62 18/29
|
Send "RV TO XY eval(x2) eval(y2) "
Wait 4
Send "SET RV.COLOR 200 200 0"
Send "SET SOUND 440"
Wait 4
Send "SET RV.COLOR 0 0 0"
```

5. Till slut ska du programmera räknaren att beräkna och visa avståndet mellan de två punkterna genom att använda avståndsformeln:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Var noggrann med parenteserna

6. Mät nu den andra ritade linjesegmentet (mellan två punkter) och jämför sedan mätningen med det beräknade värdet. Hur skiljer sig värdena åt och hur stort är det procentuella felet i mätningen. Felet beräknar du så här:

$$(mätning - avstånd) / avstånd \cdot 100$$

```
* rover62 29/29
Send "SET RV.COLOR 200 200 0"
Send "SET SOUND 440"
Wait 4
Send "SET RV.COLOR 0 0 0"

d:=√(x2-x1)²+(y2-y1)²
Disp "Avstånd=",d
EndPrgm
```

Lärarkommentar: **Wait**-kommandots värden är ungefärliga och kan justeras.

Här är det kompletta programmet:

```
Define rover62()=
Prgm
Disp "ROVER KAP6 ÖVN2"
Send "CONNECT RV"
Send "SET RV.GRID.M/UNIT .0254"
Request "X1?", x1
Request "Y1?", y1
Request "X2?", x2
Request "Y2?", y2
Text "TRYCK ENTER FÖR ATT STARTA"
Send "RV TO XY eval(x1) eval(y1)"
Wait 4
```

10 Minutes of Code

TI-Nspire-teknologi med TI-Innovator™ Rover

KAPITEL 6: ÖVNING 2

LÄRARKOMMENTARER

```
Send "SET RV.COLOR 0 200 200"
```

```
Send "SET SOUND 256"
```

```
Text "TRYCK ENTER"
```

```
Send "SET RV.COLOR 0 0 0"
```

```
Send "RV TO XY eval(x2) eval(y2)"
```

```
Wait 4
```

```
Send "SET RV.COLOR 255 0 0"
```

```
Send "SET SOUND 440"
```

```
Wait 3
```

```
Send "SET RV.COLOR 0 0 0"
```

```
 $\sqrt{(x_2-x_1)^2+(y_2-y_1)^2} \rightarrow D$ 
```

```
Disp "AVSTÅND", D
```

```
EndPrgm
```