Kapitel 6: Koordinater

Översikt: Upptäckter med slumptal kan leda till fascinerande observationer. Denna tillämpning ger dig möjlighet att utforska mer ovanliga saker om sannolikhet och att programmera Rover att flytta runt på ett rutnät.

Slumpvandring är ett experiment med datorprogrammering. Denna tillämpning knyter samman flera olika programmeringsfärdigheter.

Problemet:

Antag att din stads olika gator är upplagda i ett fyrkantigt rutmönster och att din skola ligger vid (0, 0) i rutnätet. Ditt hem är vid (7, 3), som representerar 7 kvarter öster och 3 kvarter norr om skolan. (Överväg gärna förändringar hos denna position.) Bilden till höger är en plottning av dessa två punkter.

Från skolan går du ett kvarter i slumpmässig riktning (norr, söder, öster eller väster). Vid varje korsning går du sedan ett kvarter i slumpmässig riktning igen. Kommer du någonsin att komma hem? Hur många kvarter kommer du vandra innan du kommer hem? Kommer du sluta på grund av utmattning innan du kommer hem?

Planering är en viktig del av kodningen. Tänk på vad Rover kan göra och vad ditt programmeringsspråk kan göra.

Tänk på att när man arbetar med slumptal så är vi i händerna på maskinen. Det kan ta lång tid för Rover att nå hem, så vi ska lägga till en bestämmelse om att Rover endast får förflytta sig ett begränsat antal kvarter innan du slutar.

Lärarkommentar: Förbered eleverna så att programmet kan använda en Input-sats för koordinaterna (7, 3) för hemmet. Använd variabler för den positionen från början. I själva verket är det bättre att använda variabler så mycket som möjligt (snarare än specifika värden). Användningen av variabler gör det möjligt att enkelt ändra värden eller modifiera programmet att använda Input-satser snarare än Lagra-satser för att initiera simuleringens tillstånd.

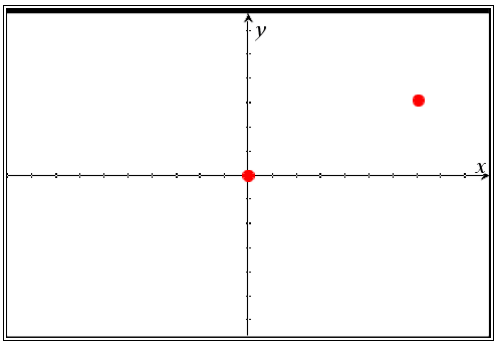
1. Starta programmet med de vanliga initiala satserna. Ställ in Rovers rutnätsstorlek till något mindre än standardvärdet 10 cm.

   Tänk på att det kommandot finns i programeditorn under **Hub > Rover (RV) > RV setup...** och det förändrar Rovers "enhet" för rörelse (används i **FORWARD 1**) från 10 cm till något annat, vilket möjliggör fler rutnätspunkter på ett mindre utrymme.

Tillämpning: Slumpvandring

Syfte:

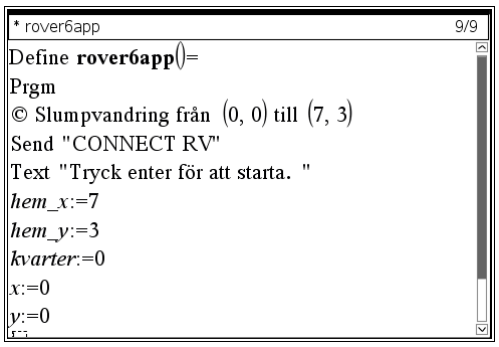
- Använda koordinatrörelse för att simulera en slumpvandring
- Använda räkneverk och ackumulatorer i ett program
- Använda sammansatta villkor med **inte** och **och**.



Initiera variabler

2. Spara hemkoordinaterna (7, 3) och startnumret för passerade kvarter (0) i variablerna **hem_x**, **hem_y** respektive **kvarter**. Variabeln **kvarter** kommer att användas för att hålla reda på antalet kvarter som Rover ”går” och kommer att användas för att stoppa programmet om Rover går för långt. Rover slutar av utmattning när den har gått ett angivet antal kvarter.

   Vi använder här 20 kvarter som ett värde för utmattningssträckan och använder variabeln *sluta* för detta värde.



3. Rover kommer hem när dess position, som vi kallar (***x***, ***y***), är (7, 3). Initiera båda variablerna, ***x*** och ***y***, att vara noll.

Huvudloopen

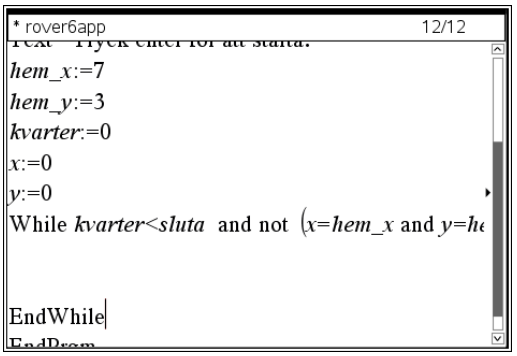
4. Huvudloopen hos programmet består av en **While**-loop med två villkor som ska adresseras. Villkoren är ”när Rover är hemma”(när x = hem_x och y= hem_y) eller slutar av utmattning (när ***kvarter*** = ***sluta***).

   While-loopen fortsätter så länge som villkoren är falska så vi ställer upp motsatta villkor:

   While kvarter<sluta and not (x=x_hem och y=hem_y)

   Kom ihåg att fasta värden som är relaterade till problemet lagras i **sluta, hem_x** och **hem_y**.

   not( ) används för att försäkra att programmet fortsätter så länge som Rover ***inte*** är ”**hemma**”. Rent logiskt är **and not** motsatsen till **or** (eller).



5. Kom ihåg att inkludera en **EndWhile**-sats om du skriver in koden ”för hand”.

Inne i loopen

6. Tillväxten av **kvarter** (antalet passerade kvarter): **kvarter:=kvarter+1**.

7. Välj riktning slumpmässigt (norr, söder, öst, eller väst):
   - Rover’s **TO ANGLE**-kommando vrider Rover till en “absolut” riktning: 0 är öst, 90 är norr, 180 är väst, och 270 är syd
   - **randInt(0,3)** ger ett slumpmässigt heltal: 0, 1, 2, eller 3.
   - Multiplicera detta värde med 90 för att få 0, 90, 180, eller 270.
   - Satsen för att få en slumpmässig **riktning** är:

   Riktning:=90·randInt(0,3)

8. Vrid Rover till vinkeln **riktning**: **Send“RV TO ANGLE eval(riktning)”**.

9. Förflytta Rover framåt 1 enhet (1 kvarter i vår simulering): **Send(“RV FORWARD 1”)**.

10. Uppdatera Rover's position i programmet:

    Om Rover går åt norr, öka då **y** med 1
    Om Rover går åt öster, öka då **x** med 1
    Om Rover går åt söder, minska då **y** med 1
    Om Rover går åt väster, minska då **x** med 1

Inkludera några **Wait-**kommandon för att hålla programmet synkroniserat med Rover´s rörelser. Vridning tar tid och att gå framåt tar tid. **Wait-**tiderna beror på vridningsvinkeln och tillryggalagd sträcka så att du kan behöva experimentera med **Wait-**värdena.

Efter loopen

Loopen avslutas på ett av två sätt:

- Om **kvarter**=**sluta**, så slutar Rover sin vandring på grund av utmattning. Visa då "Rover slutar av utmattning" på skärmen.
- I annat fall har Rover kommit "hem". Visa då meddelandet "Rover kom hem" på skärmen.
- I båda fallen ska antalet vandrade kvarter skrivas ut på skärmen.
- Kom ihåg att använda Endif i slutet av "If…Then…Else…Endif"-strukturen.

Lärarkommentarer:

Exempel på lösning: Eleverna kan testa sin kod utan en Rover. **Send**-kommandon kommer inte att göra någon skada. Den första DISP-satsen i koden nedan anger var Rover är i sitt koordinatsystem även om Rover inte är tillgänglig. Väntetiderna (**Wait**) kan behöva justeras och kan tas bort för att köra programmet snabbare utan Rover.

```
Define rover6app()=
Prgm
Send "CONNECT RV"
Send "SET RV.GRID.M/UNIT .05"
Text "Tryck enter för att starta"
hem_x:=7
hem_y:=3
kvarter:=0
sluta:=20
x:=0
y:=0
While kvarter<sluta and not(x=hem_x and y=hem_y)
kvarter:=kvarter+1
  riktning:=90*randInt(0,3)
  Send "RV TO ANGLE eval(riktning)"
  Send "RV FORWARD 1"
 Wait
  If riktning=0:x:=x+1
```

```
  If riktning=90:y:=y+1
  If riktning=180:x:=x-1
  If riktning=270:y:=y-1
  Disp x,y
EndWhile
Disp "vandrade kvarter=",kvarter
If kvarter=sluta Then
Disp "Rover slutar av utmattning!"
Else
Disp "Rover kom hem!!"
EndIf
EndPrgm
```