



Vakdidactiek informatica : praktijkgerichte seminars

Lesgevers :

Prof. A. Hoogewijs / Annick Van Daele

Vakgroep Pure wiskunde en computeralgebra

Galglaan 2

9000 Gent

Tel. : 09/2644900 – Fax. : 09/2644993 – E-mail : Albert.Hoogewijs@rug.ac.be

**Les 7 & 8: MSW Logo
Mieke Depaemelaere**

Inhoudsopgave

Hoofdstuk 1 : Een assenstelsel in Logo tekenen.

- 1.1 Het assenstelsel.
- 1.2 Het assenstelsel tekenen in LOGO.
- 1.3 De procedure ASSEN.
- 1.4 Verfijning van de code.

Hoofdstuk 2 : Functies.

- 2.1 Definitie.
- 2.2 Functiewaarden berekenen.
- 2.3 Functies tekenen.
- 2.4 De procedure CURVE.
- 2.5 De procedure PLOT

Referenties

- Logisch Logo - Auke Sikma
- <http://cage.rug.ac.be/~bh>
- <http://www.eurologo.org>
- Approaching Precalculus Mathematics Discretely – Philip G. Lewis

Hoofdstuk 1. Een assenstelsel in Logo tekenen

1.1 Het assenstelsel.

In de lessen wiskunde hebben jullie al regelmatig met een assenstelsel gewerkt, dus zouden jullie er normaalgezien vertrouwd mee moeten zijn. Toch nog even een kleine herhaling.

Om een assenstelsel te tekenen hebben we nodig:

- Een nulpunt, het punt met coördinaten $(0,0)$
- 2 assen: de horizontale as die we de X-as noemen en de verticale as die we de Y-as noemen.
- Voor elke as hebben we een schaalfactor: dit is het aantal basiseenheden die één eenheid op een as voorstelt.

Voorbeeld:

Beschouw een blad geruit papier. We kiezen ergens een nulpunt op het blad, om het gemakkelijker te maken, nemen we een snijpunt van twee lijnen. Daarna kunnen we de X-as en de Y-as tekenen (horizontaal en verticaal door ons gekozen nulpunt). Ons assenstelsel is nog niet compleet, we moeten nog de schaalfactoren op de beide assen vastleggen. Als basiseenheid nemen we de lengte van één ruitje. Als schaalfactor voor de X-as kunnen we bijvoorbeeld twee ruitjes nemen en voor de Y-as drie ruitjes. Vanaf het nulpunt te beginnen, kunnen we de assen in stukjes onderverdelen.

Voor een bijkomend voorbeeld: zie de presentatie van in de les.

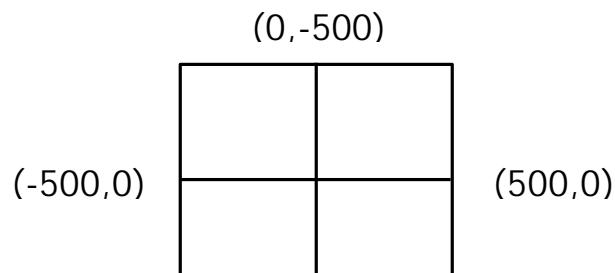
1.2 Het assenstelsel tekenen in LOGO.

De grafische wereld van Logo bevat eigenlijk ook een assenstelsel, maar het is verborgen. Als we het statusmenu bekijken zien we dat het midden van het scherm het nulpunt is. De basiseenheid in Logo is één turtlestapje en de schaalfactor is één voor beide assen.

We willen nu zelf een assenstelsel tekenen in Logo, waarin we de schaalfactor (en eventueel het nulpunt) zelf kunnen kiezen. Als nulpunt kiezen we voor de komende oefeningen telkens het nulpunt van het scherm en als schaalfactoren kiezen we bijvoorbeeld 10 turtlestapjes voor de X-as en 30 voor de Y-as.

Opmerking:

De schildpadwereld is niet oneindig groot, we kunnen geen oneindig lange lijn tekenen. De wereld is dus begrensd. In het oorspronkelijke Logo scherm kunnen we 500 turtlestapjes nemen, beginnend vanaf het nulpunt, naar boven, naar links, naar rechts of naar onderen. We kunnen punten aanduiden met coördinaten van -500 tot 500 voor x en y .



Wanneer we nu zelf een assenstelsel gaan tekenen met als nulpunt het oorspronkelijke nulpunt maar met andere schaalfactoren moeten we ook weten hoeveel eenheden we kunnen aanduiden zonder de grenzen van de wereld te verbreken.

Het aantal eenheden voor een halve as (van het nieuwe assenstelsel) is gelijk aan de halve lengte van de oorspronkelijke as in turtlestapjes gedeeld door de schaalfactor (of het aantal turtlestapjes in 1 eenheid van het nieuwe assenstelsel).

Voorbeeld:

Stel de schaalfactor voor de X-as gelijk aan 20 en de schaalfactor voor de Y-as gelijk aan 50.

De halve lengte van de oorspronkelijke as is 500 turtlestapjes.

We kunnen dus maximaal $500/20 = 25$ eenheden voorstellen op de halve X-as en $500/50 = 10$ eenheden voorstellen op de halve Y-as. Met andere woorden we kunnen punten aanduiden met coördinaten van -25 tot 25 voor x en van -10 tot 10 voor y.

Als we nu in het assenstelsel van het voorbeeld het punt (4,2) willen voorstellen, moeten we $4 * 20$ turtlestapjes in de X-richting nemen en $2 * 50$ stapjes in de Y-richting.

Om nu ons eigen assenstelsel in Logo te tekenen gaan we als volgt te werk:

Als we kijken naar een assenstelsel, dan zien we veel gelijke stukken. Ten eerste hebben we 4 halve assen, die identiek zijn, maar een andere richting ten op zichte van het nulpunt hebben. Ten tweede kunnen we die halve as ook nog eens onderverdelen per basiseenheid. Een basiseenheid kunnen we als volgt voorstellen:



We kunnen het assenstelsel dus onderverdelen op basis van de symmetrie in zulke basiseenheden. Als we vanaf het nulpunt achter elkaar in de vier richtingen zulke basiseenheden tekenen, bekomen we dus een assenstelsel.

Procedure die 1 basiseenheid tekent:

```
to basiseenheid :schaal
  fd :schaal
  local "streepje
  make "streepje :schaal / 10
  lt 90
  fd :streepje
  bk 2 * :streepje
  fd :streepje
  rt 90
end
```

We gaan eerst een afstand :schaal vooruit en dan tekenen we een verticaal (=dwars) lijntje.

Als grootte van het verticale lijntje nemen we $1/10$ van de schaalfactor en dit getal stoppen we in de lokale variabele `:streepje`.¹ Na het tekenen van het horizontale streepje, draaien we de turtle 90 graden en tekenen het verticale streepje. Op het einde draaien we de turtle terug zodat hij weer in zijn oorspronkelijke positie staat.

1.3 De procedure ASSEN.

We hebben nu een procedure `basiseenheid` die 1 eenheid tekent, deze procedure kunnen we nu gebruiken om een halve as te tekenen met een bepaalde lengte en een bepaalde schaal, bv. een halve X-as.

Procedure `halveXas`:

```
to halveXas :xaantal :xschaal
  repeat :xaantal [basiseenheid :xschaal]
  bk product :xaantal :xschaal
end
```

Deze procedure herhaalt `:xaantal` keer `basiseenheid`, er wordt dus `:xaantal` keer een basiseenheid getekend met lengte `:xschaal`. De procedure `halveYas` is volledig analoog.

```
to halveYas :yaantal :yschaal
  repeat :yaantal [basiseenheid :yschaal]
  bk product :yaantal :yschaal
end
```

Om een assenstelsel te tekenen gaan we een procedure `assen` maken met als argumenten beide schaalfactoren. `ASSEN :xschaal :yschaal`. De procedure `assen` ziet er als volgt uit :

```
to ASSEN :xschaal :yschaal
  local "xaantal
  local "yaantal
  make "xaantal int (:xdim / :xschaal)
  make "yaantal int (:ydim / :yschaal)
  repeat 2 [halveYas :yaantal :yschaal rt 90 halveXas
    :xaantal :xschaal rt 90]
end
```

De procedure maakt twee lokale variabelen aan namelijk `:xaantal` en `:yaantal` die het aantal keer bevat, dat een basiseenheid moet getekend worden op een halve as.

De variabelen `:xdim` en `:ydim` bevatten de (halve) lengte van de assen en volgende procedures veranderen de waarden van deze variabelen.

¹ lokale variabele: variabele die enkel zichtbaar is binnen de procedure waarin hij is aangemaakt. Buiten de procedure bestaat deze variabele dus niet.

```

to setXDIM :xd
    Make "xdim :xd
end

to setYDIM :yd
    Make "ydim :yd
end

```

Omdat :xschaal niet noodzakelijk een deler is van :xdim, gebruiken we het commando `int`. Dit commando geeft enkel het getal voor de komma weer. Bijvoorbeeld `int 7.6` geeft als resultaat 7.

In de laatste regel tekenen we dan de assen. Procedure `halveYas` tekent een halve Y-as, `halveXas` tekent een halve X-as. We tekenen dus eerst het positief deel van de Y-as draaien over 90 graden naar rechts en tekenen dan het positief deel van de X-as. We draaien nog eens 90 graden en staan klaar om de negatieve delen van de assen te tekenen. We kunnen op dezelfde manier te werk gaan dus we kunnen `repeat` gebruiken.

1.4 Verfining van de code.

Als we kijken naar onze procedures `halveXas` en `halveYas` dan geven we telkens respectievelijk :xschaal en :yschaal mee. We kunnen dit vermijden door in onze procedure `assen` deze variabelen globaal te maken.

De procedure `assen` wordt nu:

```

to ASSEN :xs :ys
    make "xschaal :xs
    make "yschaal :ys
    local "xaantal
    local "yaantal
    make "xaantal int (xdim / :xschaal)
    make "yaantal int (ydim / :yschaal)
    repeat 2 [halveYas :yaantal rt 90 halveXas :xaantal
rt 90]
end

```

De procedures `halveXas` en `halveYas`:

```

to halveXas :xaantal
    repeat :xaantal [basiseenheid :xschaal]
    bk product :xaantal :xschaal
end

to halveYas :yaantal
    repeat :yaantal [basiseenheid :yschaal]
    bk product :yaantal :yschaal
end

```

Nu kunnen we het assenstelsel gaan tekenen op ons scherm. We maken eerst ons scherm leeg met `cs`, we geven de variabelen `:xdim` en `:ydim` bv. de waarden 150 en 200 en dan tekenen we het assenstelsel met `assen 20 10`.

HOOFDSTUK 2 :Functies plotten in Logo.

2.1 Definitie

Een functie wordt bepaald door een functievoorschrift van de vorm: $y = f(x)$.

Voorbeeld: $y = 2x^2 + 3x - 1$

x en y zijn variabelen.

Opmerking:

Voor de functies die we gaan plotten in Logo gaan we ons beperken tot functies met twee variabelen (x en y).

2.2 Functiewaarden berekenen

Om functie waarden te evalueren of anders gezegd voor een gegeven x -waarde de corresponderende waarde van y te berekenen, hebben we het commando `run` nodig.

Beschouw het volgende voorbeeldje:

```
Make "X 7
Show Run [2 * :X + 5]
```

Dit geeft als resultaat 19. `Run` voert de instructie in de lijst uit, hier wordt de expressie $2x + 5$ geëvalueerd met x waarde 7.

Volgende procedure evalueert `:regel` (functievoorschrift onder de vorm van een lijst) voor de waarde `:x`.

```
to evalx :regel :x
  op run :regel
end
```

2.3 Functies tekenen.

Wanneer we een functie willen tekenen in een assenstelsel, bepalen we eerst de grenswaarden voor x waartussen we de functie willen plotten. Anders gezegd we kiezen de minimum en maximum waarde voor x .

Om een punt voor te stellen in ons assenstelsel moeten we, zoals eerder vermeld rekening houden met de schaalfactoren. De volgende instructie berekent de positie van een punt op ons assenstelsel rekening houdend met de schaalfactoren `:xschaal` en `:yschaal` voor een bepaalde functie `:regel`.

```
setpos list :xschaal * :x :yschaal * evalx :regel :x
```

Hoe gaan nu te werk om de curve te tekenen:

- We kiezen een eerste waarde voor x , om het gemakkelijk te maken nemen we de minimum waarde.
- Dan berekenen we $f(x)$ met x de gekozen waarde.
- Als we $f(x)$ kennen duiden we het punt met coördinaten $(x, f(x))$ aan op het assenstelsel.
- We kiezen nu een volgende x waarde, om het opnieuw gemakkelijk te maken, doen we dit door bij de oude waarde van x een constant getal op te tellen. Vb $x := x + 1$ (met 1 de constante waarde)
- We berekenen de nieuwe $f(x)$ enz....

We gaan zo verder tot de waarde van x het maximum bereikt heeft. Dan gaan we alle aangeduide punten met elkaar verbinden.

Afhankelijk van de gekozen constante waarde zal de curve vloeiender zijn.

2.4 De procedure CURVE.

Om de volledige functie af te beelden gaan we een procedure `curve` schrijven. Deze ziet er als volgt uit:

```
to curve :regel :x :max
  local "statuslijst
  make "statuslijst onthouden
  pu
  plot :regel :x :max
  pu
  herstellen :statuslijst
  pd
end
```

Deze procedure krijgt als argumenten de regel of het functievoorschrift mee, de minimum waarde voor x (dus eigenlijk de eerste x -waarde die we gaan evalueren) en de maximum waarde voor x .

Omdat we de positie van de turtle willen herstellen na het tekenen van de curve, schrijven we de procedures `onthouden` en `herstellen`. De positie van de turtle wordt bepaald door `pos` en door `heading`, deze twee gegevens slaan we op in een lijst.

```
to onthouden
  op (list pos heading)
end
```



```

to herstellen :statuslijst
  setpos first :statuslijst
  seth last :statuslijst
end

```

2.5De procedure PLOT.

De procedure `plot` moet vanaf de minimumwaarde voor `x` tot `x` groter wordt dan de maximumwaarde de functiewaarde berekenen, de turtle op deze positie zetten, de waarde voor `x` verhogen en zo de volgende positie berekenen en de berekende punten met elkaar verbinden.

We moeten dus herhaaldelijk punten berekenen en deze verbinden tot een bepaalde voorwaarde voldaan is. Met andere woorden, we kunnen met behulp van recursie de procedure `plot` schrijven.

```

to PLOT :regel :x :max
  if :x > :max [stop]
  setpos list :xschaal * :x :yschaal * evalx :regel :x
  pd
  plot :regel :x + :inc :max
end

```

De tweede lijn is de stop-voorwaarde van de recursieve procedure: er wordt gecontroleerd of de waarde `x` het maximum nog niet overschreden heeft, als dit zo is, wordt de procedure gestopt.

Dan wordt de turtle op de juiste positie gezet. De eerste keer wordt dit gedaan met `penup` (zie procedure `curve`), er wordt dus nog niets getekend.

Eens de turtle op het eerst punt staat mogen we beginnen tekenen: `pendown`.

Nu gaan we een recursieve oproep naar `plot` doen met de zelfde regel en de zelfde waarde voor `:max` maar met een andere waarde voor `:x`. We gaan de waarde van `:x` verhogen met `:inc`.

Het instellen van de waarde van `:inc` gebeurt met de procedure:

```

to SETINC :num
  make "inc :num
end

```

Het instellen van de waarde van `:inc` doen we in een afzonderlijke procedure en dus niet rechtstreeks in de procedure `plot`, zo kunnen we heel gemakkelijk de waarde ervan veranderen en het effect ervan zien.

Overzicht van de gebruikte procedures:

Naam procedure	Argumenten	Variabelen in de procedure	Betekenis
basiseenheid	<u>schaal</u>	<i>(local) streepje</i>	Tekenen van een basiseenheid met lengte <u>schaal</u> . Het dwars lijntje heeft lengte $2 * \text{streepje}$.
halveXas	<u>xaantal</u>		Tekent <u>xaantal</u> keer een basiseenheid met schaal <i>xschaal</i>
halveYas	<u>yaantal</u>		Tekent <u>yaantal</u> keer een basiseenheid met schaal <i>yschaal</i>
assen	<u>xs</u> <u>ys</u>	<i>xschaal</i> <i>yschaal</i> <i>(local) xaantal</i> <i>(local) yaantal</i>	Er worden twee globale variabelen gemaakt: <i>xschaal</i> en <i>yschaal</i> die waarden <u>xs</u> en <u>ys</u> krijgen. Vervolgens worden de lokale variabelen <i>xaantal</i> en <i>yaantal</i> berekend (bv. $\text{xaantal} = \text{int} (:x\text{dim}/:x\text{schaal})$). Tekenen van het assenstelsel.
setXdim	<u>xd</u>	<i>xdim</i>	Geeft de variabele <i>xdim</i> de waarde <u>xd</u> . Dit is de lengte van een halve X-as.
setYdim	<u>yd</u>	<i>ydim</i>	Geeft de variabele <i>ydim</i> de waarde <u>yd</u> . Dit is de lengte van een halve Y-as.
evalx	<u>regel</u> <u>x</u>		Evalueert <u>regel</u> (functievoorschrift) voor de gegeven waarde <u>x</u> .
curve	<u>regel</u> <u>x</u> <u>max</u>	<i>(local) statuslijst</i>	De gegevens van de oorspronkelijk positie worden bijgehouden in de lokale variabele <i>statuslijst</i> . Oproep van de procedure <i>plot</i> . Herstellen van de oorspronkelijke positie.
onthouden			Onthoudt de pos en de heading van de turtle in een lijst.
herstellen	<u>statuslijst</u>		Zet de turtle op positie <i>pos</i> (eerste element van <u>statuslijst</u>), en de heading van de turtle is het tweede argument van <u>statuslijst</u> .
plot	<u>regel</u> <u>x</u> <u>max</u>		Recursieve procedure die telkens een nieuwe positie bepaald, en de turtle verplaatst naar die positie tot een bepaald maximum.
setinc	<u>num</u>	<i>inc</i>	Geeft de variabele <i>inc</i> de waarde <u>num</u> . <i>inc</i> is het getal dat opgeteld wordt bij de waarde van <i>x</i> in de procedure <i>plot</i> .

