

Een inleiding tot

STARLOGO

(versie 1.2.2)

Deel 1

Algemene Inleiding

1. Inleiding

StarLogo is ontwikkeld aan het Center for Connected Learning and Computer-Based Modeling van de Tufts University. Deze versie werd ontwikkeld in het MIT Media Lab.

Het is een programmeeromgeving voor het opbouwen en ontdekken van multi-level systemen en hoort bij een klasse nieuwe 'object-based parallel modeling languages' (OBPML). Het kan gebruikt worden voor het onderzoeken van *gedecentraliseerde systemen*, zoals kudde vogels, verkeersknoopen en mierenkolonies en laat zodoende toe het gedrag van duizenden objecten tegelijk te controleren. Elk object heeft daarbij zijn eigen variabelen en eigen toestand.

StarLogo werd ontworpen met twee grote doelstellingen:

- gebruikers toestaan om krachtige, verklarende modellen van complexe systemen te ontwikkelen
- eenvoudig te gebruiken voor beginnende programmeurs en studenten

2. De spelers of 'agents'

StarLogo omvat drie types karakters:

Turtles

De inwoners van de StarLogo wereld zijn deze grafische wezens, turtles genoemd. Een turtle kan elk type object weergeven: mieren of bijen in een kolonie, auto's in een verkeersknoop, antilichamen in een immuuniteitsysteem,... . Elke turtle heeft een positie (xcor, ycor), heading, kleur, een tekenpen, een state variabele grabbed? en een variabele breed. Indien gewenst, kan men bijkomende eigenschappen toevoegen. In StarLogo kunnen de acties van duizenden turtles tegelijk gecontroleerd worden.

Patches

Dit zijn delen van de wereld waarin de turtles leven. Patches zijn echter geen passieve voorwerpen: ze kunnen eveneens commando's uitvoeren en reageren op dichtbijzijnde turtles of patches. Ze zijn opgesteld in een rooster en worden gekenmerkt door 3 state variabelen: xcor, ycor en patch-color (pc)

De waarnemer (observer)

Deze kijkt in vogelvlucht neer op de turtles en patches. Hij kan nieuwe turtles creëren en kan de activiteit van bestaande turtles en patches controleren en bekijken via monitors. Hij kan berichten sturen naar de turtles met het verzoek een reeks commando's uit te voeren (via ask-turtles)

Alle statusvariabelen (met uitzondering van xcor en ycor van de patches) kunnen gewijzigd worden via het set commando.

3. De StarLogo omgeving

Starlogo maakt gebruik van een aantal vensters:

a. StarLogo window

Dit is opgedeeld in verschillende secties

Main window of Graphics area

Dit is een (oorspronkelijk zwart) grafisch venster met *turtles* en *patches*. De patches zijn de bouwstenen waaruit de wereld is opgebouwd. Ze hebben een vaste positie (met gehele coördinaten). De turtles kunnen bewegen over het venster (hun positie kan met decimale waarden aangegeven worden). Ze kunnen één voor één met de muis verplaatst worden door slepen. Wil men alle turtles op een bepaald patch verslepen dan kan dit door Shift-klik en slepen.

Turtles worden gekenmerkt door 4 elementen:

- Who : bevat een volgnummer (toegekend bij creëren)
- Color : geeft de huidige kleur aan (tekst of nr)
- Xcor : geeft de huidige X-coördinaat
- Ycor : geeft de huidige Y-coördinaat

Het volgnummer van een turtle kan niet gewijzigd worden.

Het info-window met bovenstaande informatie (alle turtle variabelen - standaard en user-defined) verschijnt door dubbelklik op de turtle.

Patches zijn gekenmerkt door 3 elementen:

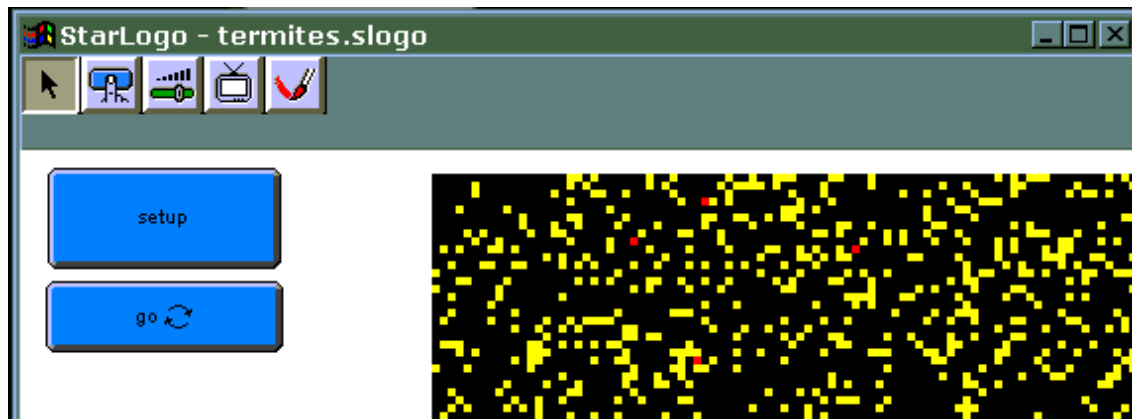
- Xcor : x-coördinaat
- Ycor : y-coördinaat
- Patch-color : geeft de huidige kleur aan (pc)

De coördinaten (zowel x- als y-coördinaat) van een patch kunnen niet gewijzigd worden, aangezien patches vaste delen zijn van het graphics window.

Het info venster bekomt men door Shift-dubbelklik op de betreffende patch.

Interface area en de toolbar

Dit bevindt zich links van de Graphics area. Dit venster bevat verschillende soorten tools die de gebruikers in staat stelt het model te runnen zoals zij dit wensen. De toolbar waarmee de tools gecreëerd worden, geeft ook een aantal bijkomende keuzes buiten het pure programmeren in het procedures window.



Er zijn 5 tools in de toolbar:

- ☞ De Arrow tool (= default waarde)
- ☞ De Button tool: creëert knoppen voor bepaalde commando's in het interface venster.
- ☞ De Slider tool: creëert slider variabelen (speciale globale variabelen)
- ☞ De Monitor tool: creëert een monitor venster. Hierdoor kan men volgen hoe bepaalde waarden tijdens het runnen veranderen.
- ☞ De Paintbrush tool: laat toe om rechtstreeks turtles en patches in het grafisch scherm te creëren

De arrow-tool

Deze wordt voorgesteld door de muis-wijzer en is standaard aangeklikt. Het creëert zelf geen interface items, maar met de arrow-tool geselecteerd kunnen alle interface items (zoals knoppen, sliders, enz.) normaal gebruikt worden.

Laat deze ingedrukt bij het runnen of bij het wijzigen van code in het procedure venster.

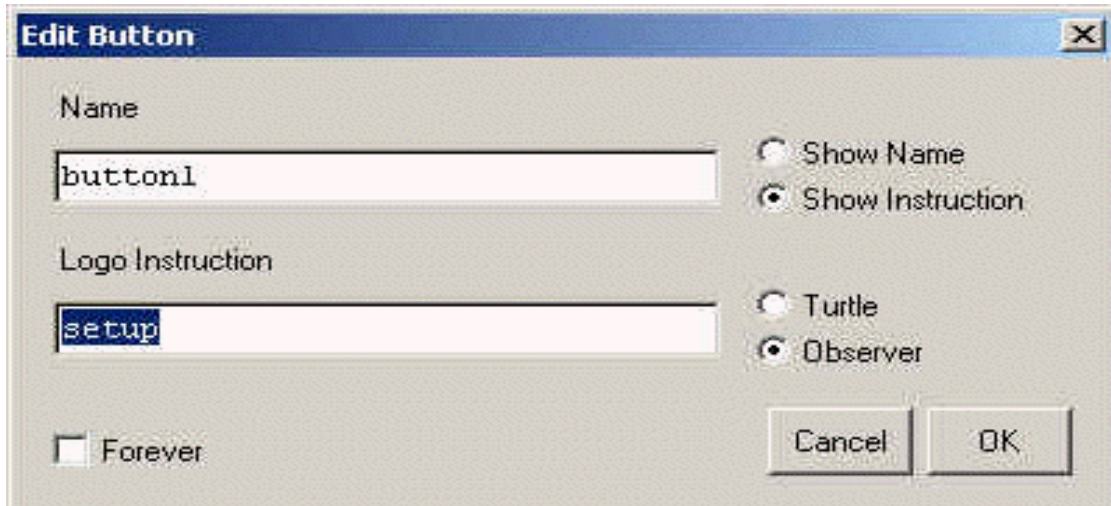
De button tool

De button wordt gebruikt om het *uitvoeren van de code te controleren*.

Er zijn 2 soorten buttons:

Once button: voert de commando's slechts één maal uit en stopt dan (=default)

Forever button: blijft de commandolijst doorlopen en herhalen tot deze actie uitgezet wordt door opnieuw aanklikken.



De commandolijst wordt opgegeven in het dialoogvenster van de button. (Selecteer via Shift-klik; open het dialoogvenster via 2 x klik):

Elke knop heeft een *naam* (bv. Setup) en bevat één of meer instructies in het *Logo instruction* venster. Veel gebruikte namen van knoppen zijn **Setup** (once-button) en de **Go** (forever-button) knop. De naam van de knop mag eventueel zelfs de instructies zelf zijn indien deze beperkt zijn tot 1 of 2)

De grootte van een knop kan gewijzigd worden:

- Selecteer de button tool in de toolbar
- Klik eenmaal op de betreffende knop
- De knop wordt omgeven door vier zwarte blokjes: verslepen van deze blokjes zorgt ervoor dat de knop van grootte gewijzigd wordt.

De slider tool

Bij het creëren van een slider wordt een nieuwe globale variabele gedefinieerd (globale variabelen zijn volledig toegankelijk t.t.z. door turtles, patches en de waarnemer). Deze variabele mag dus niet vermeld worden bij de lijst van globale variabelen aan het begin van programmacode.

Een slider is een horizontaal spoor dat de mogelijke waarden voor de variabele weergeeft. Een 'vinger' geeft de huidige waarde aan: deze kan verschoven worden naar hogere of lagere waarden. Onderaan, in een klein tekstkader, worden de naam en (nogmaals)de huidige waarde weergegeven.

Deze waarde kan gewijzigd worden op drie manieren:

- Door het verschuiven van de 'vinger' met behulp van de muis.
- Door het klikken op het spoor aan het ene of andere uiteinde.
- Door het openen van het slider's dialoogvenster en ingeven van de juiste waarde.

Zoende hebben sliders vier attributen:

- De naam : naam van de variabele die gecreëerd wordt.
- Maximum waarde: maximum waarde die de variabele kan bereiken
- Minimum waarde: minimum waarde die de variabele kan bereiken.
- Huidige waarde: waarde waarnaar de 'vinger' wijst

Samen met het creëren van een slider wordt een globale variabele gecreëerd met de zelfde naam als de slider. Bovendien worden ook nog een aantal bijkomende primitieven gecreëerd die het toelaten de waarde van de slider dynamisch te wijzigen (tijdens de run of vanuit het command center).

Bijvoorbeeld:

Stel dat er een slider met de naam 'aantal' gecreëerd werd, dan kunnen volgende primitieven aangeroepen worden:

set aantal number: met number een getal
(is number > maximumwaarde dan aantal=maximumwaarde;
is number < minimumwaarde dan wordt aantal=minimumwaarde)

set aantal-max number: de maximumwaarde van de slider wordt veranderd in number

set aantal-min number: de minimumwaarde van de slider wordt veranderd in number

set aantal-inc number: de stapgrootte wordt gewijzigd in number

De grootte van de slider zelf kan, in tegenstelling tot buttons en monitors, niet gewijzigd worden.

De monitor tool

Dit is een interface object dat voortdurend een bepaalde uitdrukking evalueert en het resultaat weergeeft (enkel numerieke waarden).

Bij het creëren geeft men de uit te voeren instructie op en ook een bepaalde graad van precisie op (tussen 0 en 3 decimalen).

De instructie kan een variabele zijn, een uitdrukking of een procedure die eindigt in de 'output' primitieve. Let op: monitors worden slechts periodisch geupdated

Monitors kunnen, zoals buttons, van grootte gewijzigd worden.

De paintbrush tool

De paintbrush tool wordt rechtstreeks in het grafisch venster gebruikt, zodat de omgeving naar wens kan aangepast worden: patch kleuren kunnen gewijzigd, turtles kunnen gecreëerd worden.

Bij een klik op het paintbrush tool verschijnen twee andere werkbalken:

- één voor de kleurkeuze en
- één voor het soort borstel dat men wenst te gebruiken.

Kleur: bovenaan krijgt men 2 keuzes

Een vierkant (voor het wijzigen van de patches)

Een turtle (om turtles te creëren)

Daaronder krijgt men het volledige kleurenpalet: de verticale strip links geeft de hoofdkleur aan; de rechter strip geeft in detail alle mogelijkheden binnen dit hoofdkleur

Vorm: deze werkbalk lijkt analoog aan de werkbalk in een grafisch pakket

Er zijn 12 verschillende knoppen

Penseel: om een turtle te tekenen of de kleur van een patch te wijzigen naar de huidige kleur

- Gom: (enkel voor patches) zet de kleur van de aangeklikte patch terug op zwart
- Lijn: Selecteer één punt (door klik), ga naar een tweede punt en klik daar om een lijn tussen beide punten te maken.
- Druppelteller: (enkel voor patches) wijzigt de huidige kleur naar een andere kleur door klikken op een patch van dat kleur
- Vierkant - niet gevuld
- Vierkant - gevuld
- Ovaal - niet gevuld
- Ovaal - gevuld
- Polygoon
- Verfpot: kleurt alle patches met hetzelfde kleur als de aangeklikte naar het huidige kleur (voor patches)
Creëert een turtle van de huidige kleur in al deze patches

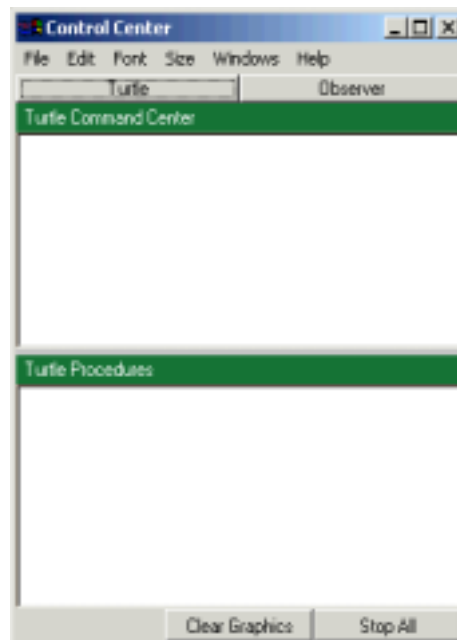
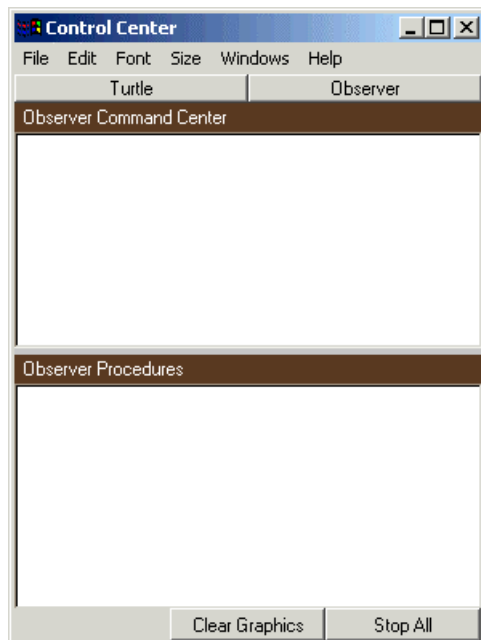
b. Control center

In het Command Center deel kunnen StarLogo commando's ingegeven worden (zie ook Logo). Een commando kan opnieuw gerund worden door de cursor opnieuw naar deze lijn te brengen en Return te duwen.

De Procedures area bevat de programma-code.

Hier worden alle non-slider variabelen gecreëerd en de procedures uitgeschreven.

Er wordt een onderscheid gemaakt tussen Turtle en Observer commando's.



Elk programma is steeds opgedeeld in verschillende procedures (observer- en turtle-).

Elke procedure begint met het woord TO en eindigt met END.

De procedures kunnen opgeroepen worden vanuit het resp. *Command Center*, via een *knop* of via een *andere procedure*.

De andere vensters zijn te bereiken via de optie *Windows* in het Control Center

c. Output window

Dit wordt gebruikt voor het printen en weergeven van data gegenereerd door StarLogo. Het PRINT commando print in dit venster

d. Plot window

Hier kunnen real-time grafieken gecreëerd worden, tijdens het runnen van het project.

e. Info window

Dit venster is bedoeld voor bijkomende uitleg, commentaar en instructies

4. Het StarLogo menu

In de Menu Bar komen verschillende mogelijkheden aan bod.

Het File Menu laat elementaire file-bewerkingen toe (bewaren, openen,...).

Het Windows menu laat toe vensters te openen die niet automatisch bij het opstarten van het project geopend werden.

Het Edit Menu bevat een aantal items die de algemene parameters van een project kunnen beïnvloeden.

5. Voorbeelden

zie voorbeelden uitgewerkt tijdens de les

Deel 2

Basiscommando's en procedures

1. Inleiding - StarLogo commando's

Hieronder volgt een overzicht van veelgebruikte commando's en procedures in StarLogo.

Er zijn 3 soorten karakters in de Starlogo wereld: turtles, patches en observer. Elke soort heeft zijn specifieke commando's.

De **observer** controleert de patches en de creatie van turtles. Hij kan, zoals reeds vermeld, boodschappen sturen naar turtles en patches met het verzoek een opdracht uit te voeren.

Wil je gebruik deze commando's in het **observer command center**

Turtles creëren	Create-turtles number Crt number
Scherf wissen	Clearall : wist alles (turtles en patches) , (ook ca) Clearturtles: wist enkel turtles , (ook ct) Cleargraphics: wist enkel patches, (ook cg)
Turtles opdracht geven	Ask-turtles [opdrachten]
Patches opdracht geven	Ask-patches[opdrachten]

De **turtles** kunnen rondlopen en allerlei opdrachten uitvoeren. Ze kunnen zelfs patches beïnvloeden.

Wil je gebruik deze commando's in her **turtle command center**

Turtles laten bewegen	Fd step
Veranderen van richting	Rt angle Lt angle Seth direction (met direction tussen 0 en 360)
Tekenen gedurende bewegen	Pd : tekent het gevolgde pad (penup) Pu : tekent het pad niet (pendown)
Kleur wijzigen	Setc newcolor : geeft de turtle nieuwe kleur
Turtle op bepaalde positie zetten	Setxy a b: met x-coördinaat a en y-coördinaat b

2. StarLogo variabelen

Bij het begin van elk programma moeten de bijkomende variabelen voor elke 'agent' gedeclareerd worden.

a) Soorten variabelen

Er zijn drie types variabelen in StarLogo: turtle variabelen, patch variabelen en observer variabelen (ook globale variabelen genoemd).

Je kan nieuwe 'state'-variabelen creëren voor turtles, patches en de observer. Het creëren van een variabele gaat namelijk gepaard met het creëren van nieuwe procedures die het gebruik van de variabele vergemakkelijken.

Alle niet-lokale variabelen moeten bovenaan het Procedures Window gedeclareerd worden, gegroepeerd per soort, tussen vierkante haken.

Je kan deze commando's in het turtle procedure window of in het observer procedure window zetten, maar niet in beide!

Turtles-own [age speed]

Patches-own [food]

Globals [time]

De compiler onderscheidt variabelen door de spaties ertussen: elke variabele moet een reeks aaneengesloten symbolen zijn. Er mogen naast alfanumerieke tekens ook andere symbolen gebruikt worden. Er wordt geen onderscheid gemaakt tussen hoofd- en kleine letters.

De naam van een variabele moet **UNIEK** zijn. Dit is de enige beperking in de keuze van variabelen.

Een aantal symbolen zijn niet toegestaan in de naam van variabelen zoals) ([en]. Dit zou bij de compiler verwarring teweeg brengen.

In bovenstaand voorbeeld worden de variabelen age en speed gecreëerd voor elke turtle, de variabele food voor elke patch.

Sliders zijn globale variabelen die niet bovenaan in het Procedures Window moeten gedeclareerd worden (Dit zou zelfs resulteren in een foutmelding). Ze worden op exact dezelfde manier gebruikt als andere globals. Sliders hebben een paar bijkomende procedures voor het aangeven en wijzigen van hun grenzen (zie hoger).

b) Creëren van variabelen

Bij het creëren van een variabele worden een aantal bijkomende commando's door StarLogo gecreëerd.

De waarden van een andere turtle of patch kunnen altijd gewijzigd of opgevraagd worden door gebruik te maken van volgende suffixen:

-at	heeft als argument een dx en dy relatief t.o.v. de caller
-of	heeft als argument een ID nummer of de turtle of patch
-towards	heeft als argument een hoek (relatief t.o.v. heading van de caller) en een afstand in turtlestapjes

We nemen bijvoorbeeld de turtle-variabele age

Turtles-own [age]

Na het creëren zijn volgende commando's mogelijk:

set age waarde om de waarde van age van de turtle te wijzigen
age geeft de waarde van de turtles age
set age age + increment verhoogt de waarde van age met increment

om de waarden van andere turtles op te vragen:

age-at xcor ycor geeft de waarde van age van alle turtles op relatieve afstand xcor ycor
age-of turtle id-nr geeft de waarde van age van de turtle met opgegeven ID-nummer
age-towards angel distance: geeft lijst van alle turtels met waarden van age variabele die op afstand distance zijn op relatieve hoek angle

Om ze te wijzigen gebruikt men het set commando:

setage-at, setage-of en setage-towards

Wil je gebruik deze commando's

Iets uitvoeren voor een turtle wiens age > waarde	If age > check [commands] Enkel die turtles wiens age > check voeren de commando's uit
De variabele speed aanpassen voor turtles die voldoen aan bepaalde voorwaarde	If condition [set speed speed * multiplier] Die turtles die voldoen aan de voorwaarde, verhogen hun speed multiplier keer

Voor een patch-variabele 'food' zijn volgende commando's beschikbaar:

food rapporteert de waarde van 'food' (!! Patch-commando)
setfood de patch geeft zijn eigen 'food' een bepaalde waarde

food-at rapporteert de waarde van 'food' van een patch op enige afstand
food-towards rapporteert de waarde van 'food' van de patch op afstand distance onder hoek angle

Om de waarden van patches op afstand te wijzigen gebruikt men opnieuw (analoog als bij turtles): setfood-at, setfood-towards

Voor een observer-variabele 'time' die vermeld werd bij de globale variabelen, zijn beschikbaar:

Time geeft de waarde van 'time'
settime de waarnemer geeft 'time' een waarde

Voor een slider 'nummer' (zie ook deel1):

nummer geeft de waarde van 'nummer'
nummer-max geeft de waarde van 'nummer's huidige maximum
nummer-min geeft de waarde van 'nummer's huidige minimum
number-inc geeft de huidige waarde van het increment

setnummer	observer geeft 'nummer' een waarde
setnummer-max	observer stelt de huidige maximum-waarde van 'nummer' in
setnummer-min	observer stelt de huidige minimum-waarde van 'nummer' in
setnummer-inc	observer stelt de waarde in van het increment

c) Lokale variabelen

Deze zijn, zoals globals, niet verbonden aan een bepaalde turtle of patch. Ze bestaan echter alleen in de procedure waarbinnen ze gedeclareerd werd.

Lokale variabelen hebben namen die beginnen met dubbelepunt.

let :variabele waarde

Toekennen van een waarde:

set variabele waarde

d) Parameters

Wanneer men een procedure voor het berekenen van één of andere waarde, meerdere keren wenst te gebruiken, met andere beginwaarden, dan kan men deze beginwaarden telkens meegeven als parameter.

Parameters bestaan, zoals lokale variabelen, enkel in de procedure waarmee ze geassocieerd worden. Zijn er meerdere parameters, dan worden deze gescheiden door spaties.

e) StarLogo breeds

Turtles kunnen verdeeld (geclasserd) worden in verschillende klassen.

Een 'breed' is in essentie een nieuwe turtle-variabele.

Het creëren gebeurt zoals bij andere variabelen, door een opsomming bij het begin van de code in het observer of het turtle procedures window

```
breeds [lambs frogs birds]
```

Door het creëren van breeds creëert men een reeks commando's die ermee gepaard gaan, zoals bijvoorbeeld:

Ask-frogs [list-of-commands] : de observer wacht tot alle frog-turtles klaar zijn met de commando's alvorens verder te gaan

Count-frogs : telt alle frogs

Count-frogs-here : telst alle frogs op de huidige patch

Setbreed lambs : turtles zetten hun breed op lambs

3. Programmastructuren en turtle interacties

Volgende structuren controleren het verloop van een programma

Wil je	gebruik dan
Commando uitvoeren afhankelijk van voorwaarde	If condition [statement] Statement wordt enkel uitgevoerd als voorwaarde voldaan is. If not (voorwaarde) [dothis] Dothis wordt slechts uitgevoerd als voorwaarde niet waar is
Commando's herhalen	Repeat times [commands] De commando's worden times keer herhaald
Andere turtles op bepaalde patch?	Ifelse count-turtles-here > 1 [statements 1] [statements 2] zijn er nog turtles op huidige patch dan [statements 1], zoniet [statements2]
Variabel van andere turtle wijzigen	Set variabele-at x y value (zie ook hoger) Gebruik je xcor en ycor als coördinaten, dan wordt de variabele van alle turtles op huidige patch gewijzigd!

4. Monitor commando's

Bij het creëren van monitors, om het verloop van een programma beter te kunnen volgen, kan men volgende commando's gebruiken:

Count-color kleur	Telt het aantal turtles met die kleur
Count-turtles	Telt het totaal aantal turtles
Count-turtles-with [condition]	Telt het aantal turtles die voldoen aan aan voorwaarde

5. StarLogo procedures

a) Algemeen

In het Procedure venster worden alle non-slider variabelen gedeclareerd en alle procedures voor het project uitgeschreven.

Elke procedure heeft een naam, die gebruikt wordt bij het aanroepen.

Bijvoorbeeld:

to go

```

    fd 1          ;;moves all turtles
    seth (random 360) ; all turtles get new heading
end

```

Bij elke aanroep 'go' zal de procedure doorlopen worden voor alle turtles: elke turtle gaat 1 stap vooruit en krijgt daarna een nieuwe heading.

Elke procedure begint dus met 'to <procedure-name>': hierdoor weet de compiler dat hetgeen volgt een procedure is.

Elke procedure eindigt met 'end': de compiler beschouwt dit als het einde van de procedure.

Commentaar begint met een puntkomma: alles wat hierop volgt, tot het einde van de lijn, wordt door de compiler genegeerd

b) Oproepen

Procedures kunnen op verschillende wijzen opgeroepen worden:

- Vanuit het Command Center (door intypen van de procedurenaam)
- Vanaf een button of een monitor (in de Instruction line moet de naam van de procedure vermeld worden)
- Vanuit een andere procedure

Meestal gebruikt men één 'hoofd' -procedure (als een forever-button) die een reeks andere procedures oproept.

c) Recursie

In StarLogo kan recursief gewerkt worden (waarbij een procedure zichzelf oproept).

Er moet vooral op gelet worden oneindige loops te vermijden: er moet een voorwaarde getest worden die, indien waar, als resultaat geeft dat uit de procedure gesprongen wordt via de 'stop' primitieve.

Een voorbeeld:

```

to recursively-move-to-edges
  ifelse ( (xcor >= screen-edge) or
           (xcor <= (0 - screen-edge)) or
           (ycor >= screen-edge) or
           (ycor <= (0 - screen-edge)))
    [stop]
    [fd 1
     recursively-move-to-edges]
end

```

Deze routine zorgt ervoor dat elke turtle naar de rand toe beweegt. Afhankelijk van elke turtles heading en positie zal het minder lang of langer duren eer deze de rand bereikt, maar aan het einde bereiken alle turtles de rand.

Alle verdere informatie over Starlogo voor PC's kan je vinden op

<http://www.media.mit.edu/starlogo/>

Hier zijn ook een aantal uitgewerkte projecten te bekijken. De voorbeelden uit de les vind je op onze website terug.