

# Functies, expressies en controlestructuren

Dieter Claeys

## Vorige les

- Object

Wat is een object?

Wat zijn de objecten in deze wereld?



## Vorige les (2)

- Object
- Instructies (of methodes, of procedures)  
Geef enkele voorbeelden van instructies voor de spinrobot in onderstaande wereld.



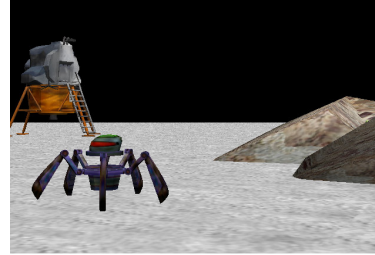
## Vorige les (3)

- Object
- Instructies (of methodes, of procedures)
- Do in order
- Do together  
Welke voorbeelden hebben we de vorige les gezien i.v.m. Do in order en Do together?

## Deze les

- Functies
- Expressies
- Controlestructuren: if/else

## Probleem



- Laat de spinrobot naar de rots wandelen.
- Als methode gebruik je move forward. Maar wat geef je mee als argument voor de afstand?

## Oplossing

- Eerste methode: via **trial and error**

## Oplossing

- Eerste methode: via trial en error
- Tweede methode: vraag de afstand aan Alice

## Functies

- Niet alle eigenschappen van een object zijn beschikbaar in de properties lijst.
- Beschikbare eigenschappen: color, opacity, ...
- Probleem: hoogte, breedte, positie, afstand tot een ander object?
- Dit kun je te weten komen door dit te vragen aan Alice. Zo een vraag heet een **FUNCTIE**

## Data types

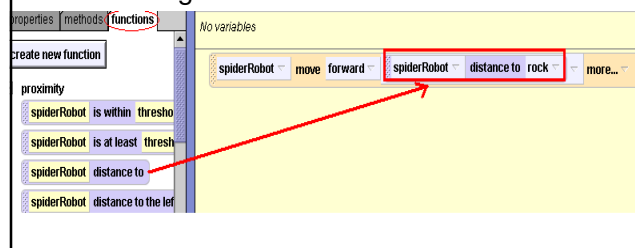
- Antwoord op vraag is de waarde van de eigenschap waar je naar vroeg. Hieronder staan een aantal veelvoorkomende **types** van waarden:
  - **Number**: 5
  - **Boolean**: true of false
  - **String**: "hallo"
  - **Object**: een spinrobot

## Oplossing

- Eerste methode: via trial en error
- Tweede methode: via een FUNCTIE aan Alice vragen

## Oplossing

- Eerste methode: via trial en error
- Tweede methode: via een FUNCTIE aan Alice vragen

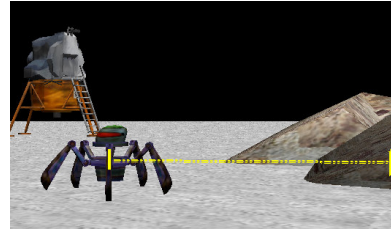


## Oplossing

- Eerste methode: via trial en error
- Tweede methode: via een FUNCTIE aan Alice vragen
- **Bug**: botsing → belang van testen  
Hoe komt dit???

## Reden bug

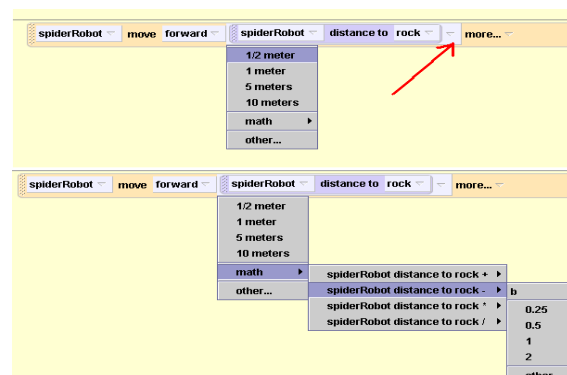
- Afstand wordt gemeten tussen het middenpunt van beide objecten



## Expressies

- Eigenlijk moet je de halve breedte van de rots en van de spinrobot aftrekken van de lengte die de spinrobot loopt.
- Dit wordt een **EXPRESSIE** genoemd
- Vbden:
  - Lengte1 + lengte2
  - Lengte \* breedte

## Expressies (2)

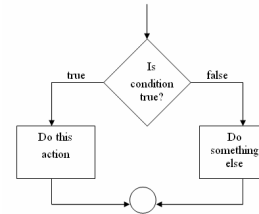


## Probleem

- Er is een alien op wielen verstopt achter de rotsen.
- De spinrobot komt bij de rots aan en zoekt de alien. Is het nodig dat de spinrobot zijn nek omhoog steekt om over de rots te kijken? Wat bepaalt of het nodig is of de spinrobot zijn nek uitsteekt?

## Controle structuren: If/else

- Hangt af van een beslissing



## Oplossing probleem

- Conditie: als de spinrobot kleiner is dan de rots
- Dan: doe de nek 0.2 meter omhoog en daarna weer 0.2 meter omlaag
- Anders: doe niets

## Probleem

- Hoe drukken we uit dat de spinrobot's lengte kleiner moet zijn dan de rotshoogte?
- → RELATIONELE OPERATORS
- Vbden:
  - kleur == rood
  - Lengte >=2 meter
- Te vinden in functions van de world

**world's details**

properties methods **functions**

**math**

- `a == b` *is equal to*
- `a != b` *is not equal to*
- `a > b` *is greater than*
- `a >= b` *is greater than or equal to*
- `a < b` *is less than*
- `a <= b` *is less than or equal to*

### Voorbeeld

**World's details**

properties methods **functions**

- `a != b`
- `a > b`
- `a >= b`
- `a < b`
- `a <= b`

**World.my first method**

No variables

```

if (a < b)
  (Do Nothing)
else
  (Do Nothing)
  
```

A red arrow points from the `a < b` block in the 'World's details' functions list to the `a < b` condition in the 'World.my first method' code block.

### Oefening

- Open wereld politiewagen
- Laat de gele auto 15 meter vooruit rijden
- Zorg ervoor dat als de auto het stopbord passeerde, dat de politiewagen in de richting van de auto rijdt. Anders doet de politiewagen niets.
- Test nu nog eens hetzelfde scenario in het geval dat de auto 5 meter vooruit rijdt.
- Extra: Laat de auto vooruit rijden maar zorg ervoor dat hij aan het stopbord stopt.

### Volgende les

- Nieuwe controlestructuur: loop
- Oefeningen