Herhaling

- Programmeren is het samen plaatsen van onderdelen
- Er bestaan 4 fundamentele stukken
  - Instructie
  - Controle structuur
  - Functie
  - Expressie

Instructie

- Een instructie wordt uitgevoerd tijdens het uitvoeren van het programma (play)
- In object geörienteerde talen heten instructies methodes
- Voorbeeld : move, turn to face

Controle Structuur

Een controle structuur bepaalt de volgorde van instructies

- 3 Types
  - Sequentie / simultaan : Do Together / Do in order
  - Voorwaardelijk : if / else
  - Herhaling : loop / while

Functie

- Een functie stelt een vraag
- In Alice gebruiken we ze om informatie te controleren over
  - Eigenschappen van objecten
  - De relatie tussen objecten
  - Huidige condities

Functie

- Er wordt geantwoord in de vorm van een value (waarde)
- Types van values
  - Booleaans
  - Nummer
  - String
  - Object
  - Coördinaat
  - .....

Expressie

- Een expressie is een wiskundige bewerking op nummers of andere waarden
- Gebruikte operatoren
  - Wiskundige operatoren (+,-,x,/)
  - Relationele operatoren (==,<,>,…)

Hoofdstuk 4 (1)

Methodes

Grote programma's

- Wanneer je beter wordt in programma's schrijven, merk je dat programma's zeer snel een grote hoeveelheid regels zullen worden
- Spelletjes en andere echte software bestaan uit duizenden tot miljoenen regels code
- Nood aan ordening

```
//-------------------------------------------------------------------------------------
//  Purpose  : This function calculates a Type C direct-acting PID controller.
//             This function should be called once every Ts seconds (Ts = 5 here)
//  Variables used: (all variables are doubles (Floating point)
//          xk : The process variable PV (= measured HLT temperature, read from ADC)
// pid_output : The output of the PID controller (a value between 0% and 100 %)
//        tset : The setpoint value for the HLT temperature
// pid_enabled: Release signal: 1 = Start control, 0 = disable PID controller
//
// Constants used: c1, c2, Ts, Kc, Ti, Td
//-------------------------------------------------------------------------------------
double ek;   // the error tern
double xk_1; // previous value of xk
double lpf, lpf_1, lpf_2; //LPF output and previous values of LPF output

  // Calculate the error tern (Setpoint value - Process variable (actual HLT temperature)
ek  = tset - xk;

 // Loupass filter for D term, input is x[k] instead of e[k] (type C PID controller)
lpf = c1 * lpf_1 + c2 * (xk + xk_1);

if (pid_enabled)
{
     // P term is only dependent of x[k] and not e[k] (type C PID controller)
    P_term = -Kc * (xk - xk_1);

    I_term = (Kc * Ts / Ti) * ek; // I term is still dependent of e[k]

    // D term is only dependent of filtered version of x[k]
    D_term = (-Kc * Td / Ts) * (lpf - 2 * lpf_1 + lpf_2);

    // Add all the parts to the previous value of the controller output,
    // making it a PID velocity algorithm.
    pid_output = pid_output + P_term + I_term + D_term;
}
else
{   // PID controller is not enabled, set all terms and the output to zero
    P_term = I_term = D_term = pid_output = 0.0;
}

// Now update the previous values of the signals
xk_1  = xk;    // PV[k-1] = PV[k], the actual HLT temperature
lpf_2 = lpf_1; // previous values of the loupass filter output, the filtered version of xk
lpf_1 = lpf;   //

// Last thing to do is to limit the controller output to a maximum and minimum value
if (pid_output > 100.0) // more than 100 %?
{
   pid_output = 100.0;  // yes, limit to 100 %
}
else if (pid_output < 0.0) // less than 0 %?
{
   pid_output = 0.0; // yes, limit to 0 %
}
```

Methode

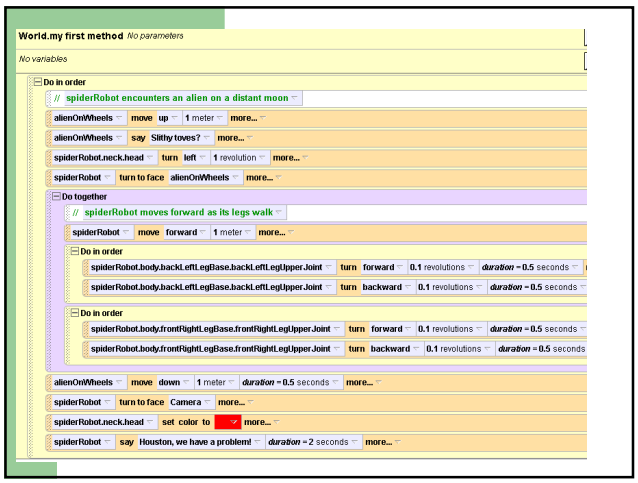
- Is een gecoördineerde sequentie instructies
- Een methode verdeelt je code in kleinere, beter beheerbare stukken
- Analogie met een boek : hoofdstukken

Abstractie

- Voorbeeld rondlopen herhaald:
  - Loop 2 rondjes rond een vierkant
  - = maak 8x de beweging vooruit en kwartdraai
  - = maak 2x een rondje
- Maak de methode ‘rondje’
- Methode ‘rondje’ is een abstractie
- Abstracties laten toe om over een complexere handeling te denken als een basishandeling

Oplossing grote programma’s

- Een mogelijk oplossing voor de vele lijnen code is deze opsplitsen in kleinere methoden
- Voorbeeld uit de vorige les : FirstEncounter
- We maken een mogelijk verdeling hiervoor: verras, onderzoek en reageer



Opsplitsing

Do in order

verras – spiderRobot en alienOnWheels verrassen elkaar

onderzoek – spiderRobot gaat alienOnWheels onderzoeken

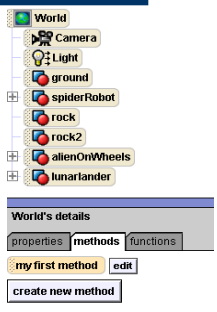
reageer– alienOnWheels verstopt zich en spiderRobot stuurt een bericht

- Het opbreken van de algemene taak in kleinere taken heten we stapsgewijs verfijnen
- Dit kan herhaald worden

verras
Do in order
alienOnWheels moves up
alienOnWheels says "Slithy toves?"
spiderRobot's head turns around

Wereld-niveau methode

- Wereld-niveau methodes zijn methodes waarin objecten met elkaar interageren
- verras,onderzoek en reageer zijn wereld-niveau methodes
- Voeg deze toe aan de wereld

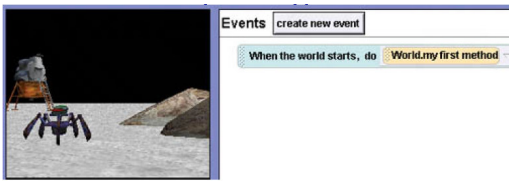


Methodes aanroepen

- Een gemaakte methode doet niets op zichzelf
- De methode moet aangeroepen worden
- Alice start de methode World.my first method bij het starten van de wereld
- Andere methoden kunnen in deze methode aangeroepen worden

Andere manier tot aanroepen

- Events kunnen ook methoden aanroepen
- Verander het onderstaande event en roep de gewenste methode aan



Klasse

- Een klasse bepaalt een groep gelijksoortige objecten
- Objecten in eenzelfde klasse hebben een aantal methodes gemeenschappelijk
- Voorbeelden van klassen
  - People, Building, Environment, Animal
- Naam van een klasse begint met een hoofdletter

Klasse

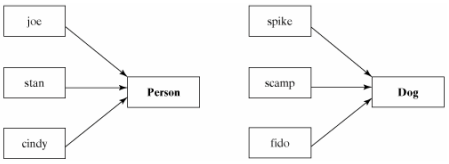
Een specifiek object in een klasse heet een instantie van de klasse

Een instantie begint met een kleine letter, net als een methode

- oscar
- fido

Klasse

- Objecten in dezelfde klasse hebben een gemeenschappelijke structuur
- Voorbeeld: alle honden hebben vier poten, alle mensen hebben 2 benen



Klassenmethodes

- Als een methode slechts één object iets laat uitvoeren
- Methode kan toevoegd worden aan het object
- Methoden die toegevoegd kunnen worden aan een hond :
  - kwispelen,
  - Blaffen
  - Lopen
  - Eigen staart achtervolgen
  - …