

Methods of Development

College 2

“Imperatief programmeren”

Arjan Scherpenisse

arjan.scherpenisse@kmt.hku.nl

@acscherp

Deze week

- De Opdracht
- Fractals en algorithmes
- Pseudocode
- Scratch

Huiswerk review

- Bijna iedereen ingeleverd :-D
- Leuk om te bewaren en later terug te zien
- Iedereen zelfde beoordeling



```
package |
import flash.display.*;
import flash.events.*;

public class Nio extends MovieClip {

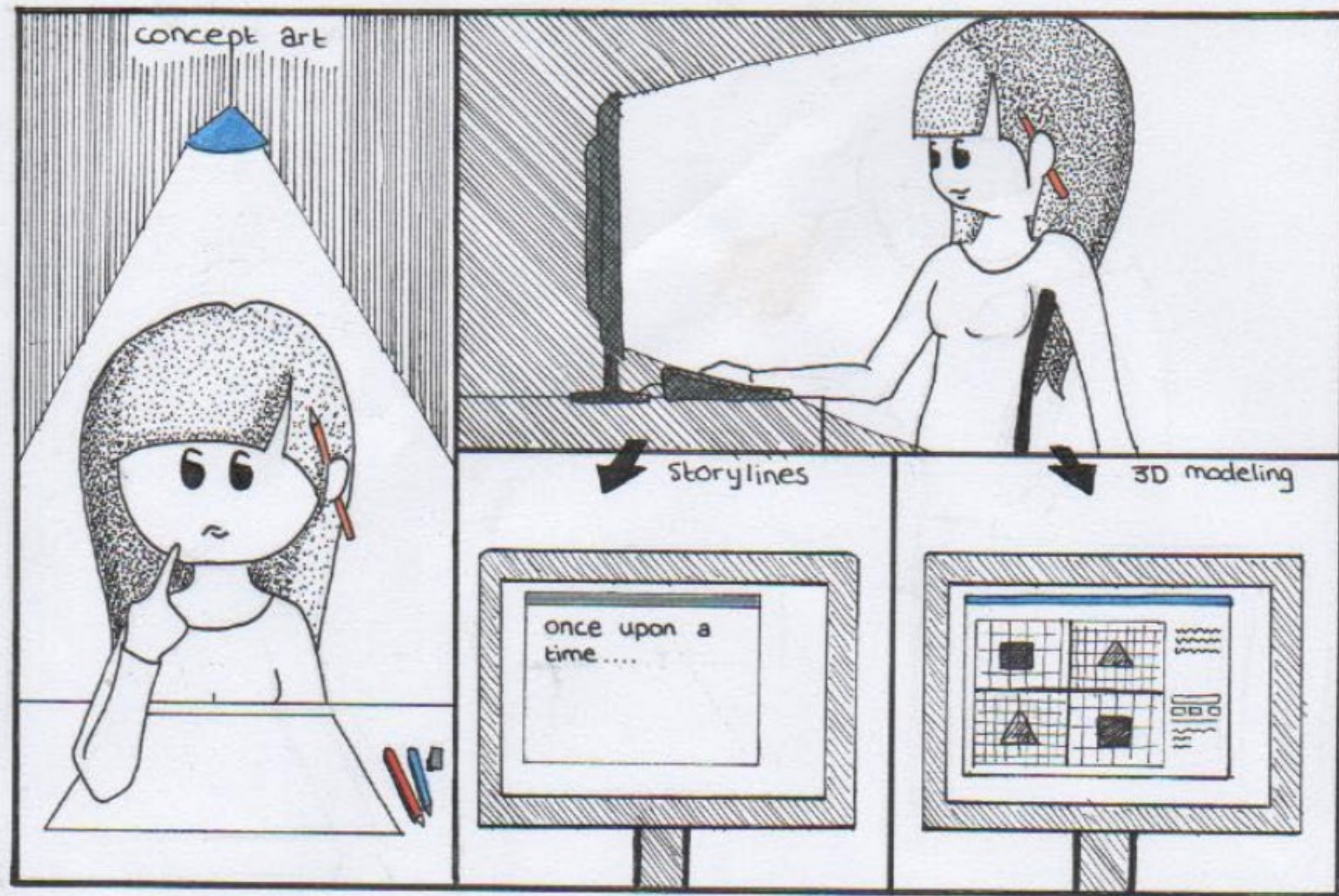
    public var woord:String="";
    public var waarde:Number=20;
    public var bewegingx:Number=0;
    public var bewegingy:Number=0;
    public var positionx:Number;
    public var positiony:Number;
    public var rotatiesnelheid:Number=0;
    public var aantal:Number=0;

    public function Nio() {
        addEventListener(Event.ENTER_FRAME,doelets);
    }
}
```



*I program for a hardworking team of professionals
Who share the same passion*

steven honders



Tk zie mijzelf later, werkend bij een klein bedrijfje dat
bijvoorbeeld indiegames produceerd. Hierbij is mijn rol
het tot stand brengen van een concept en het
uitwerken ervan. Hopelijk heeft deze rol
te maken met het opzetten van storylines
en characterdesign. -Denise ☺



yu ting chiu

Indy Dev



Adam Op ten Berg 

@Indy Amsterdam, Netherlands

I work at a little indy game company and make games!

 Following



Tweets

Favorites

Following ▾

Followers ▾

Lists ▾



@Derpo Alex Veens

Love the new game series! Keep m' coming! 8U



44 min ago



@Honk Jochem De Koning

YES can't wait for the new release of your new game!
Hope it will be as good as the other 2 :P...

2 days ago



@Grasmat Bas Adriaanse

I have to admit... Im addicted to your games! XD
Please stop making them! lolz



4 days ago



@Boom4lolz Kenrick Ouwehand

Thats a very nice game you have there. Would be a shame
if something happened to it...



4 days ago



- Wereld veranderen thomas versteeg
- Mannetje van alles julien ranzijn

```
ixion:~> php -a  
Interactive shell
```

```
php > $x = base64_decode('SWsgd2lsIGxhdGVyLCBhbHMgaWsgZ3Jvb3QgYmVuLCBncmFhZyBwcm9ncmFtbWV1ciB3b3JkZ  
W4uIEhldCBsaWVmc3QgYmlqIGVlbiBrbGVpbmUgaW5kaWUtZGV2ZWxvcGVyLiBPb2sgem91IGlrIGVyZyBncmFhZyBpbiBtaWpu  
IHZyaWplIHRpamQgZWlnZW4sIGtsZWluZSBzcgVsbGVuIHdpcGxlbjBtYWtlbiEgSWsgZ2EgbnUgYmVnaW5uZW4gbWV0IGhldCB  
sZXJlbiB2YW4gTFVBIC0gZGl0IGdhIGlrIG9vayBkb2VuIHZvb3IgbWlqbiBJU0EgLSBlbiBkYW4gd2lsIGlrIGRhYXJuYSBkZS  
BvdmVyc3RhcCBnYWFuIG1ha2VuIG5hYXIgQysrLiBPb2sgd2lsIGlrIGVlbiBiZWV0amUgZ29lZCB3b3JkZW4gaW4gcGl4ZWxhc  
nQsIG1hYXIgZGF0IGhlZWZ0IGdlZW4gcHJpb3JpdGVpdC4='');
```

```
php > echo $x;
```

```
Ik wil later, als ik groot ben, graag programmeur worden. Het liefst bij een kleine indie-developer  
. Ook zou ik erg graag in mijn vrije tijd eigen, kleine spellen willen maken! Ik ga nu beginnen met  
het leren van LUA - dit ga ik ook doen voor mijn ISA - en dan wil ik daarna de overstap gaan maken  
naar C++. Ook wil ik een beetje goed worden in pixelart, maar dat heeft geen prioriteit.
```

```
php > █
```

peter dijkstra

Literatuurlijst

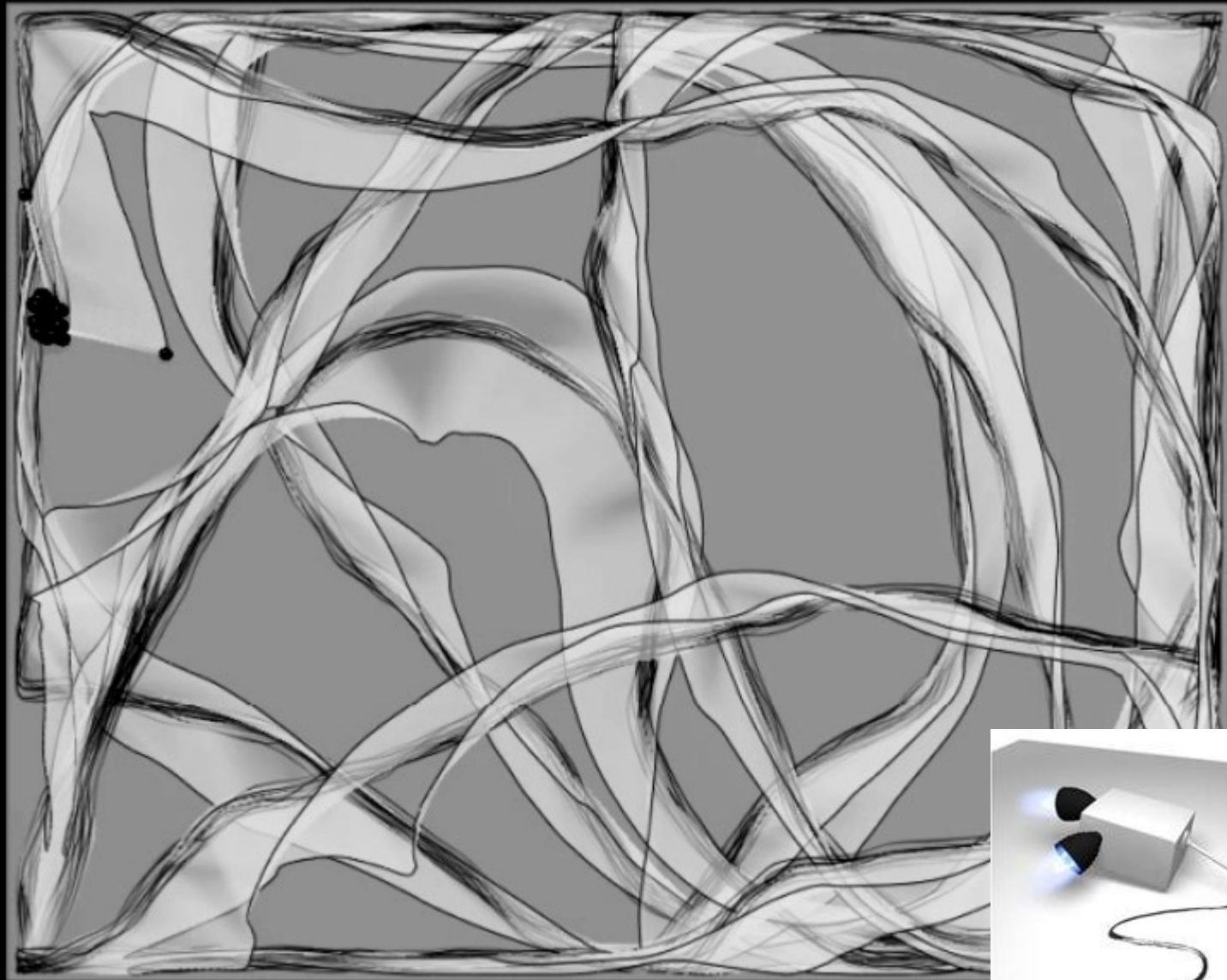
- Software studies (*Matthew Fuller*)
 - o.a. artikelen “CVS”, “Variable”
- Code Complete
(*2nd Ed*) by *Steve McConnell*
- Software engineering (*Ian Sommerville*)

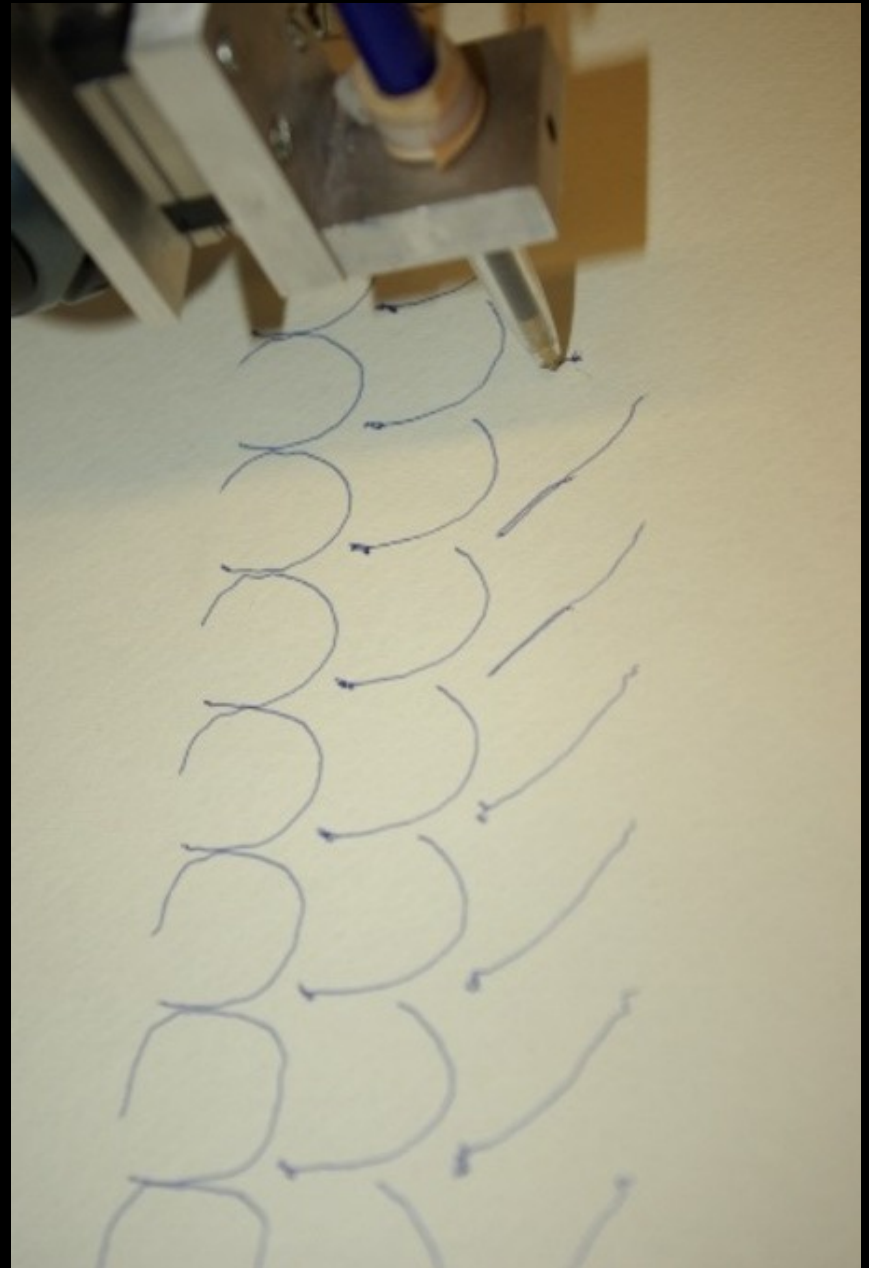
“Methods of Development”...?

- == *thinking* of development



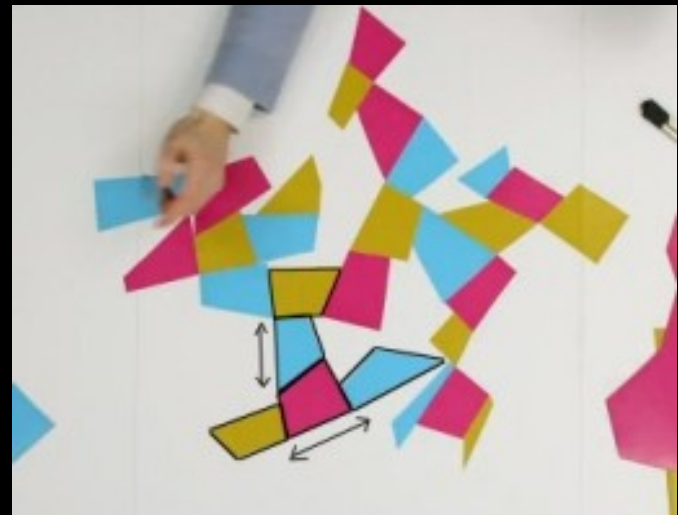
De programmeur als vormgever





“Conditional design”

- Art initiative
- Graphic design through *process* and *logic*.
- Broccoli, Convex city plan





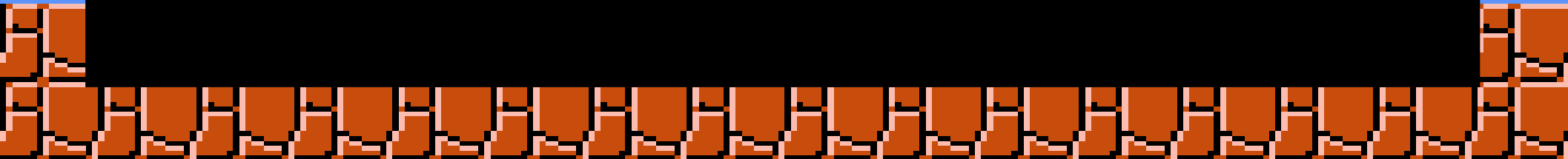
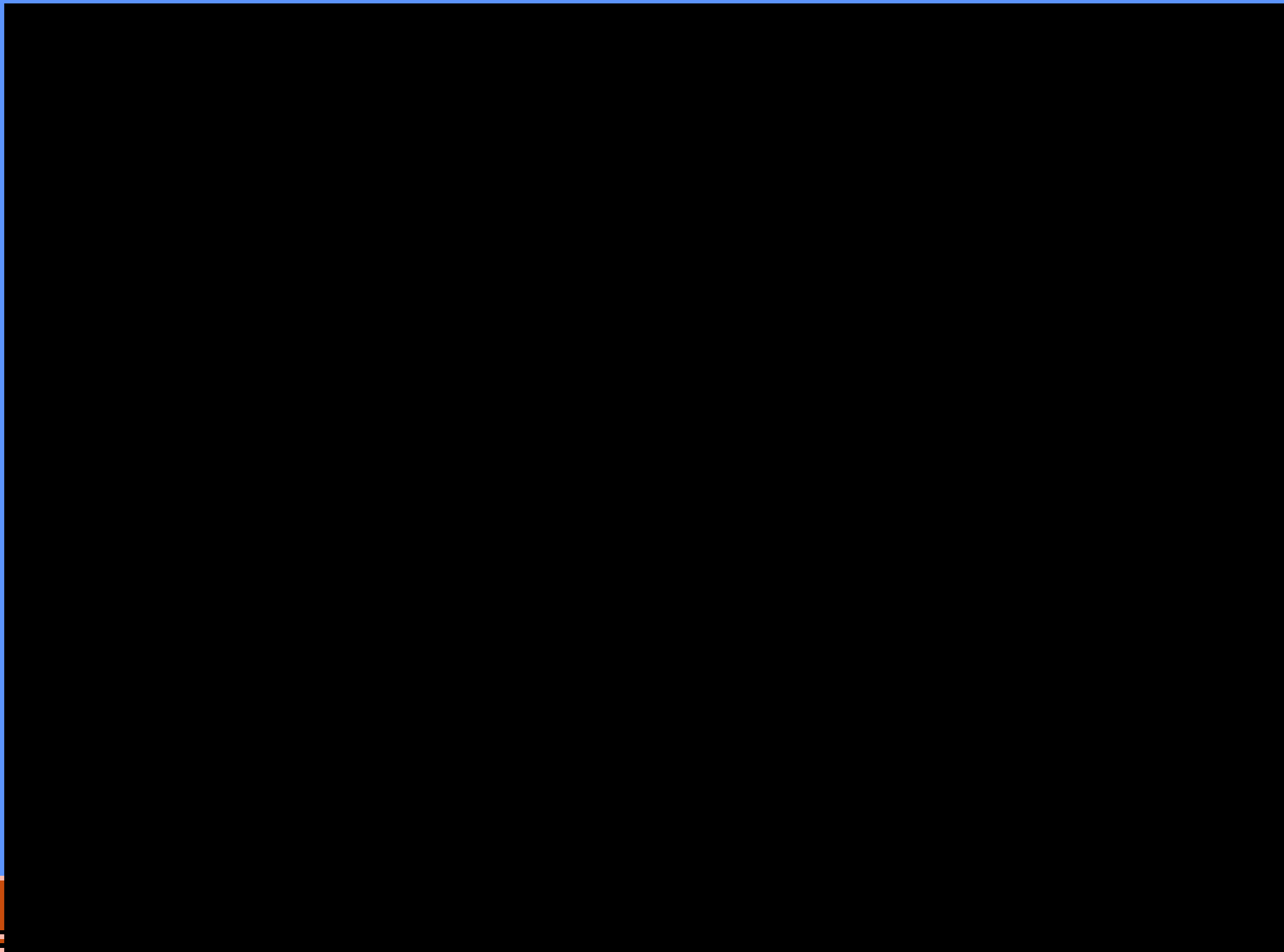
Computer-gegenereerd

- Fractals
 - Visuele uitingen van logische regels
 - Procedural textures, meshes
- Genetisch / evolutionair
 - Simulatie van de natuur: simpele regels, complex systeem
 - Games zoals *spore*



“Genetica is the tool to use
when you absolutely must
kick booty.”

Käy Vriend
Texture Artist

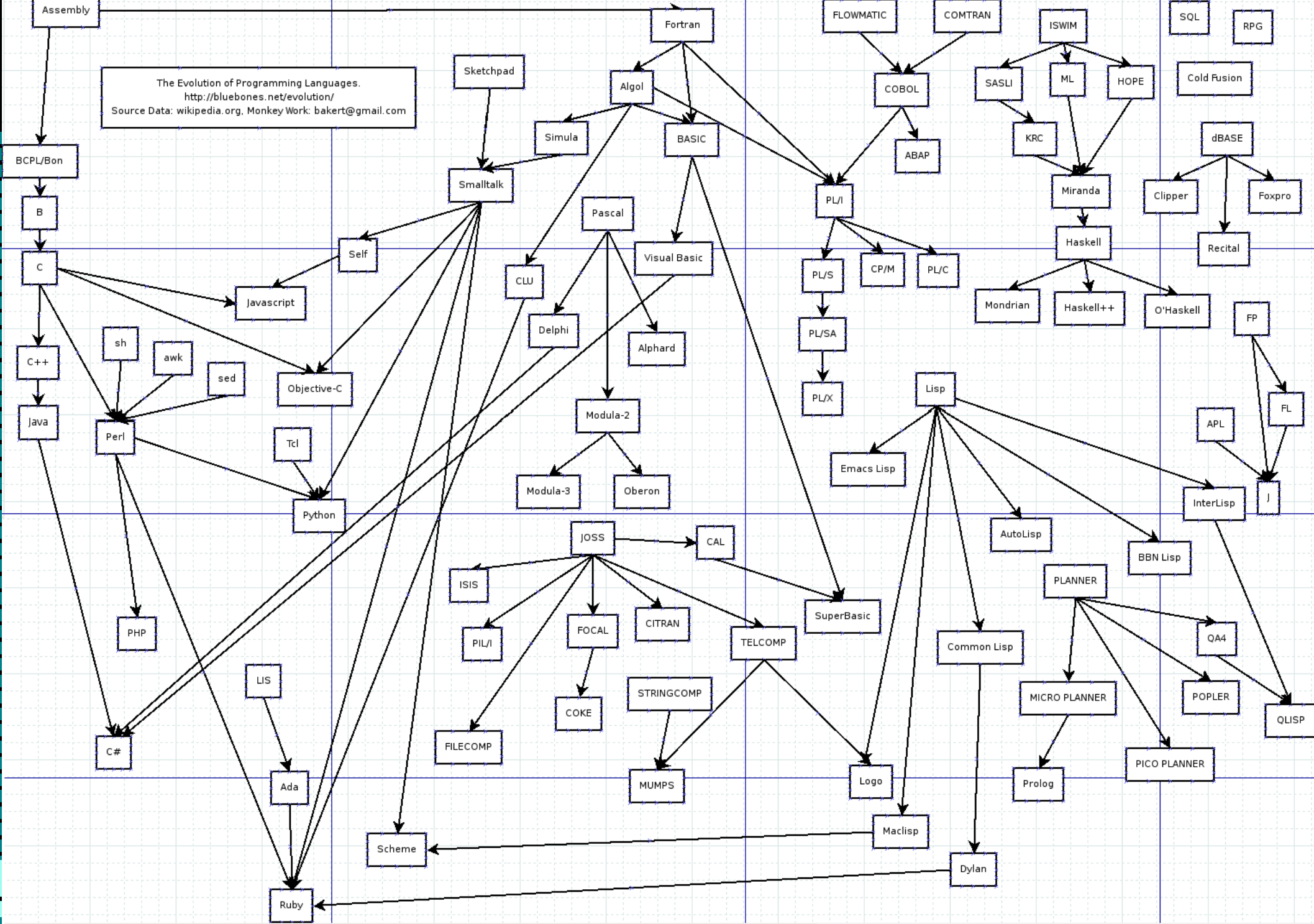


Programmeren

- Het aansturen van een computer via instructies
- Deze instructies vormen de *broncode* van het programma
- Instructies meestal worden niet direct in *machinetaal* geschreven
- *Compilers* en *interpreters* maken van broncode machinetaal

Programmeertalen

- Er zijn vele duizenden programmeertalen
 - Bekende: Python, Ruby, PHP, C, C++,
ActionScript, ...
 - Exotische: Shakespeare, BrainFuck, Chef, ...
- Instructies worden achter elkaar (sequentieel) uitgevoerd



Maar hoe ziet code er dan uit?


```
%%=====
%% API
%%=====

%% @spec start_link(Args) -> {ok,Pid} | ignore | {error,Error}
%% @doc Starts the server
start_link(Args) when is_list(Args) ->
    gen_server:start_link(?MODULE, Args, []).

reload(Context) ->
    z_notifier:notify(development_reload, Context).

make(Context) ->
    z_notifier:notify(development_make, Context).

%% @doc Stream specific debug information to an area on the current page.
debug_stream(TargetId, What, Context) ->
    z_notifier:notify1(#debug_stream{target=TargetId, what=What}, Context).

%% @doc Stream all debug information of a certain kind to the target id on the user agent.
observe_debug_stream(#debug_stream{target=TargetId, what=What}, Context) ->
    start_debug_stream(TargetId, What, Context).

pid_observe_development_reload(Pid, development_reload, _Context) ->
    gen_server:cast(Pid, development_reload).

pid_observe_development_make(Pid, development_make, _Context) ->
    gen_server:cast(Pid, development_make).

%% @doc Called when a file is changed on disk. Decides what to do.
%% @spec file_changed(string()) -> ok
file_changed(Verb, F) ->
    case file_blacklisted(F) of
        true -> nop;
        false ->
            case handle_file(Verb, filename:extension(F), F) of
                undefined -> ok;
                Message -> send_message(os:type(), z_string:trim(Message))
            end
    end
end,
ok.
```

Javascript

```
$widget("ui.inlineLightbox",
{
  _init: function() {
    var self = this;

    self.parentElement = self.element.parents(self.options.parent).first();

    self.offset = self.parentElement.offset();
    self.width = self.parentElement.width();

    self.img = $("img:first", self.element);
    self.aspect = self.img.width() / self.img.height();

    self.element.click(function(e) {
      self.clicked(e);
    });

    if (self.element.attr("id")) {
      if (document.location.hash == "#" + self.element.attr("id")) {
        self.clicked();
      }
    }
    self.zoomIcon = $("").attr("src", "/lib/images/icon_zoom.png").hide();

    self.element.mouseover(function() {
      var o = self.img.offset();

      var po = self.parentElement.offset();
      var pad = (self.img.outerWidth()-self.img.width())/2;
      self.zoomIcon
        .appendTo(self.parentElement)
        .css({position: "absolute", top: o.top-po.top+pad-1, left: self.img.outerWidth()+o.left-po.left-pad-self.zoomIcon.wid
        .show();
    })
    .mouseout(function() {
      self.zoomIcon.hide();
    });
  },

  clicked: function(e) {
    var self = this;

    if (e) {
      e.preventDefault();
    }

    if (self.element.attr("id")) {
      if (document.location.hash != "#" + self.element.attr("id")) {
        var id = self.element.attr("id");
        self.element.attr("id", "");
      }
    }
  }
}
```

...**WOW.**

- Gelukkig ga ik jullie niet leren code schrijven :-)
- Wat we wel doen is *pseudo-code* maken.

Pseudocode

- “Pseudocode wordt gebruikt om *algoritmen* vast te leggen op een door mensen leesbare manier met behoud van de stappen.” (wikipedia)

Kletskommen van Holtkamp

- Verwarm oven voor op 220°C. Roer de boter, basterdsuiker en melk door elkaar. Meng kaneel, zout, amandelen erdoor, en daarna de bloem.
- Doe het beslag in een spuitzak en spuit het in kleine bolletjes op een bakplaat
- Bak de kletskommen in slechts enkele minuten in de oven. Leg ze daarna op het aanrecht om af te koelen

verwarm de oven voor op 220°

REPEAT

roer ingredienten door elkaar

UNTIL egaal mengsel

REPEAT

sprit bolletje op bakplaat

UNTIL bakplaat vol OF deeg op

IF er is nog deeg **THEN**

bewaar rest van deeg in koelkast

END

zet plaat in oven

wacht 3 minuten

haal plaat uit oven

Pseudocode: Statements

- “Roer de ingrediënten door elkaar”
- “Verwarm de oven voor”
- *Geparametriseerd:*
 - “Verwarm de oven voor op 220° graden”
 - “Verwarm de oven voor op X graden”

Pseudocode: Condities

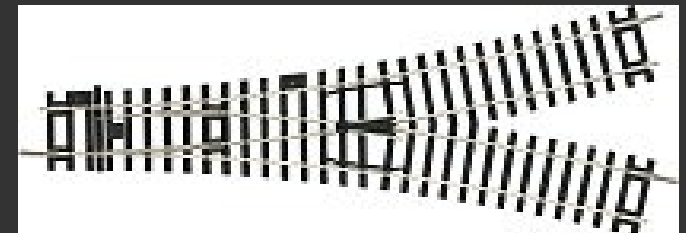
- Oorzaak → gevolg: “Als dit, dan dat”
- IF iets, THEN doewat, (ELSE doewatanders)

Expressie

Statement(s)

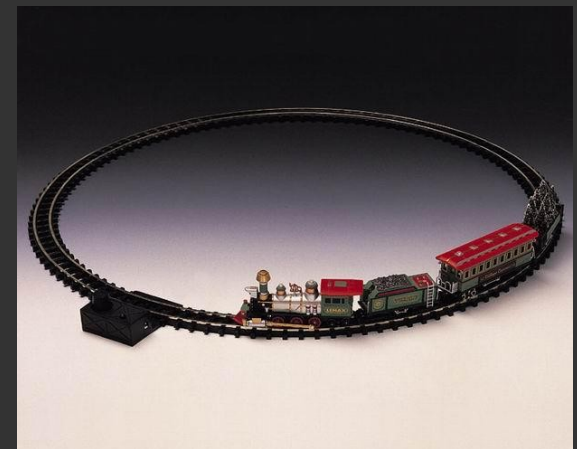
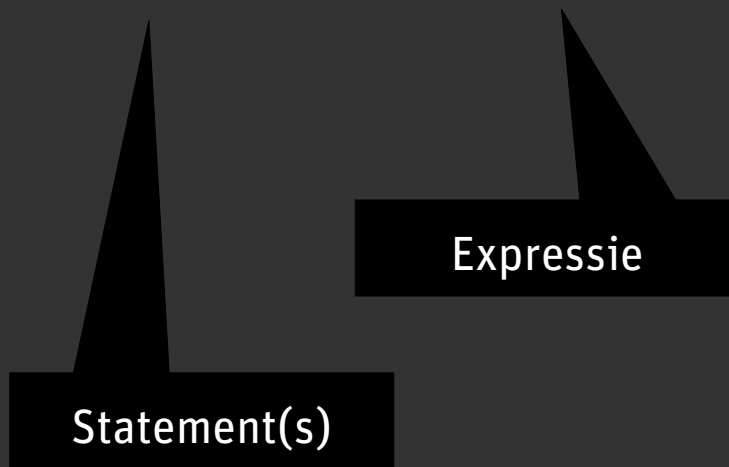
Statement(s)

- IF oventemperatuur te hoog
THEN stop met voorverwarmen



Pseudocode: Loops

- “Doe hetzelfde, totdat er iets geldig is”
- REPEAT **doewat** UNTIL **iets** (*geldig is*)



Pseudocode: Variabelen

- Een “vakje” waar iets in kan worden bewaard

SET *gewensteTemperatuur* TO 220

activeer oven

REPEAT

 SET *huidigeTemperatuur* TO de huidige oventemperatuur

UNTIL *huidigeTemperatuur* >= *gewensteTemperatuur*

Pseudocode: Expressies

- “`huidigeTemperatuur GROTER DAN gewensteTemperatuur`”
→ is een expressie
- Een “check”.
- Expressies zijn altijd “waar” (**true**) of “niet waar” (**false**)
- Worden gebruikt in condities (**IF..THEN**)
- Worden gebruikt in loops (**REPEAT..UNTIL**)

Pseudocode richtlijnen

- Duidelijke namen van dingen
- Syntax niet zo belangrijk
 - ...maar wees wel *consistent!*
- Balans abstractie / specificiteit

Balletje-balletje, te simpel:

Initialize some stuff

REPEAT

Get user input

UNTIL user choose the right cup

Show victory screen

Balletje-balletje, te complex:

```
BOOLEAN Match
INTEGER UserCup
INTEGER RandomCup

PRINT "GUESS THE CUP (C) 2011 HKU PRODUCTIONS, ALL RIGHTS RESERVED"

PRINT "Which cup do you think the ball is under?"
PRINT "Press 1 for cup number one"
PRINT "Press 2 for cup number two"
PRINT "Press 3 for cup number three"

SET NumberOfCups to 3

CALL RandomNumberGenerator with NumberOfCups RETURNING RandomCup
REPEAT

  IF KeyPressed EQUALS 1 THEN
    SET UserCup to 1
  ELSE IF KeyPressed EQUALS 2 THEN
    SET UserCup to 2
  ELSE IF KeyPressed EQUALS 3 THEN
    SET UserCup to 3

  SET Match to UserCup EQUALS RandomCup

  IF Match NOT EQUALS true
    PRINT "The ball was not under the cup of your choice, please try again"

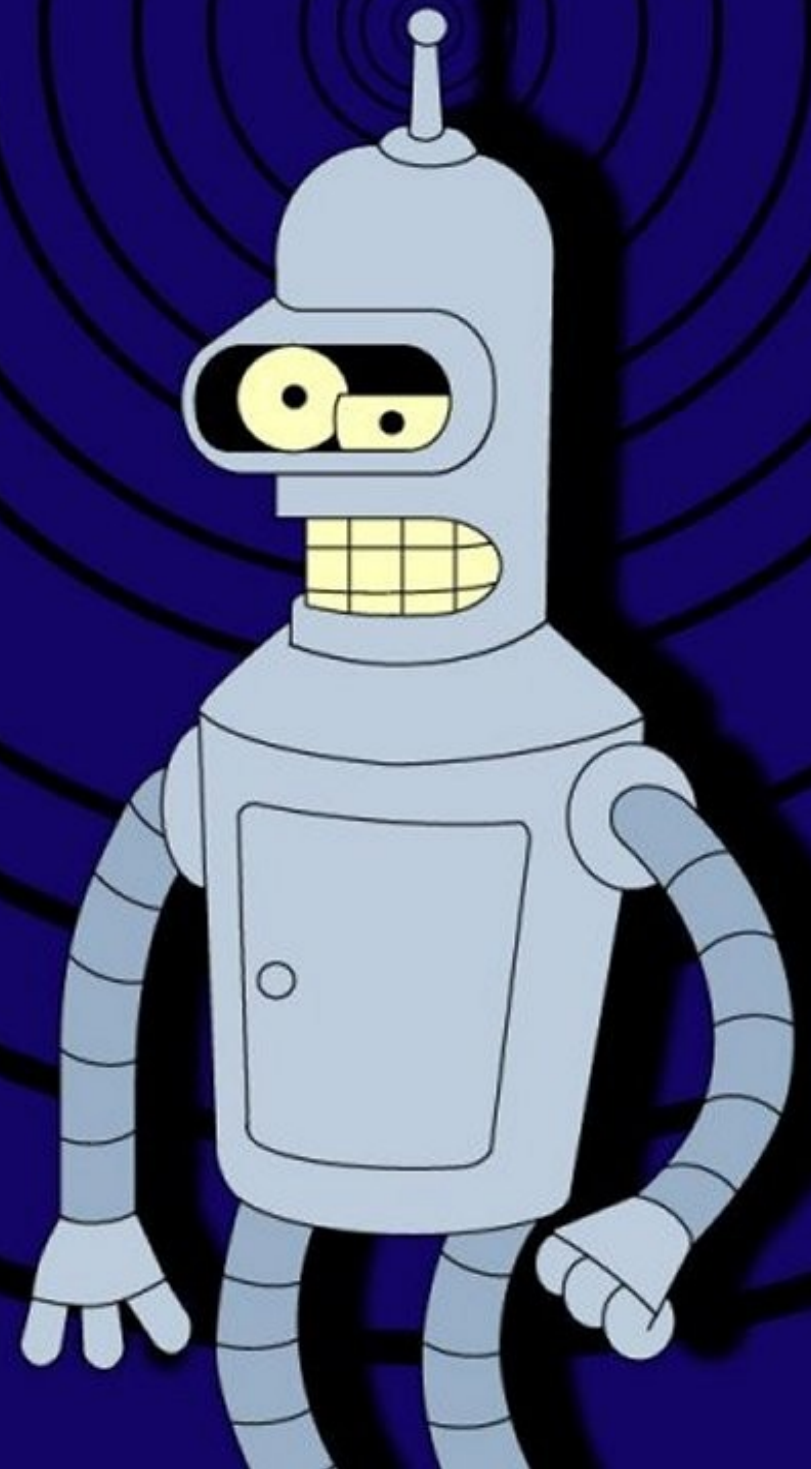
UNTIL Match EQUALS true

PRINT "Congratulations! You have guessed the right cup!"
PRINT "Please play again!"
```

Oefenopdracht:

- Ga een dagelijkse bezigheid van jezelf na, en schrijf deze op in pseudocode. Probeer het juiste niveau van detail te vinden; verlies jezelf niet in details, maar blijf ook niet op een te hoog niveau.
Verzin bezigheid waarbij je minimaal 1 variabele, 1 loop en 1 if-then-else nodig hebt.

Zijn we er nog?



THE TERRAPIN LOGO LANGUAGE

WRITTEN BY L. KLOTZ, P. SOBALVARRO
AND S. HAIN UNDER THE SUPERVISION
OF H. ABELSON.

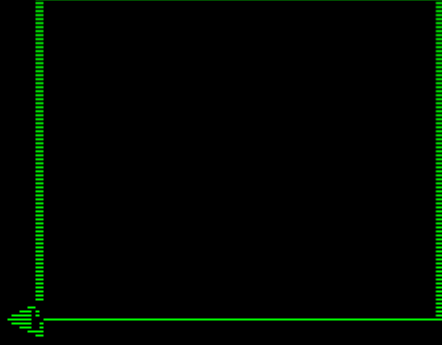
COPYRIGHT (C) 1981 MIT
VERSION 1.0

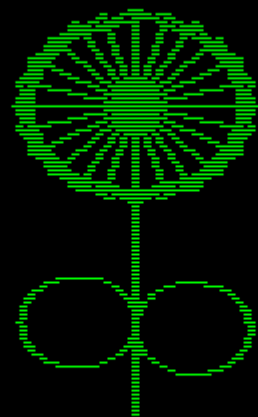
WELCOME TO LOGO
? 



TI
CORPORATION

www.ti.com
SOS
SOS





?

SCRATCH



SCRATCH



File Edit Share Help



- Motion
- Control
- Looks
- Sensing
- Sound
- Operators
- Pen
- Variables



Sprite 1
x: 70 y: 0 direction: 90

Scripts Costumes Sounds

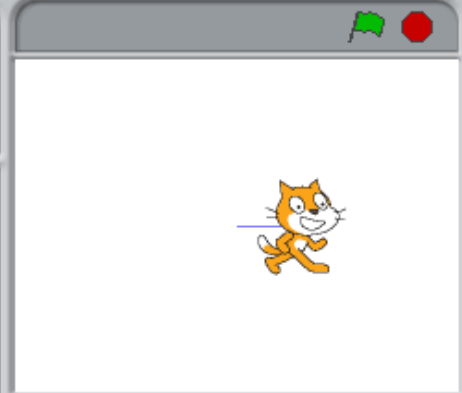
- clear
- pen down
- pen up
- set pen color to
- change pen color by 10
- set pen color to 0
- change pen shade by 10
- set pen shade to 50
- change pen size by 1
- set pen size to 1
- stamp

pen down

pen up

move 10 steps

repeat until



x: -1486 y: -1024

Sprite 1

Stage

Komende colleges

- ~~College 1: waar hebben we het over~~
- ~~College 2: imperatief programmeren~~
- College 3: object-orientatie
- College 4: vervolg OO, eindopdracht
- Tentamen (begin december)

Opdracht:

- Experimenteer met *Scratch* om een schetsopzet te maken voor een kleine (arcade) game. Het hoeft niet compleet af maar het moet wel iets doen! Het gaat om het script: graphics/sound zijn leuk maar tellen niet zwaar mee.
- Jullie hebben hier 2 weken voor.
- **Deadline: dinsdag 22 november 23:59u**
- arjan.scherpenisse@kmt.hku.nl

Tot volgende week!



Methods of Development

College 2

“Imperatief programmeren”

Arjan Scherpenisse

arjan.scherpenisse@kmt.hku.nl

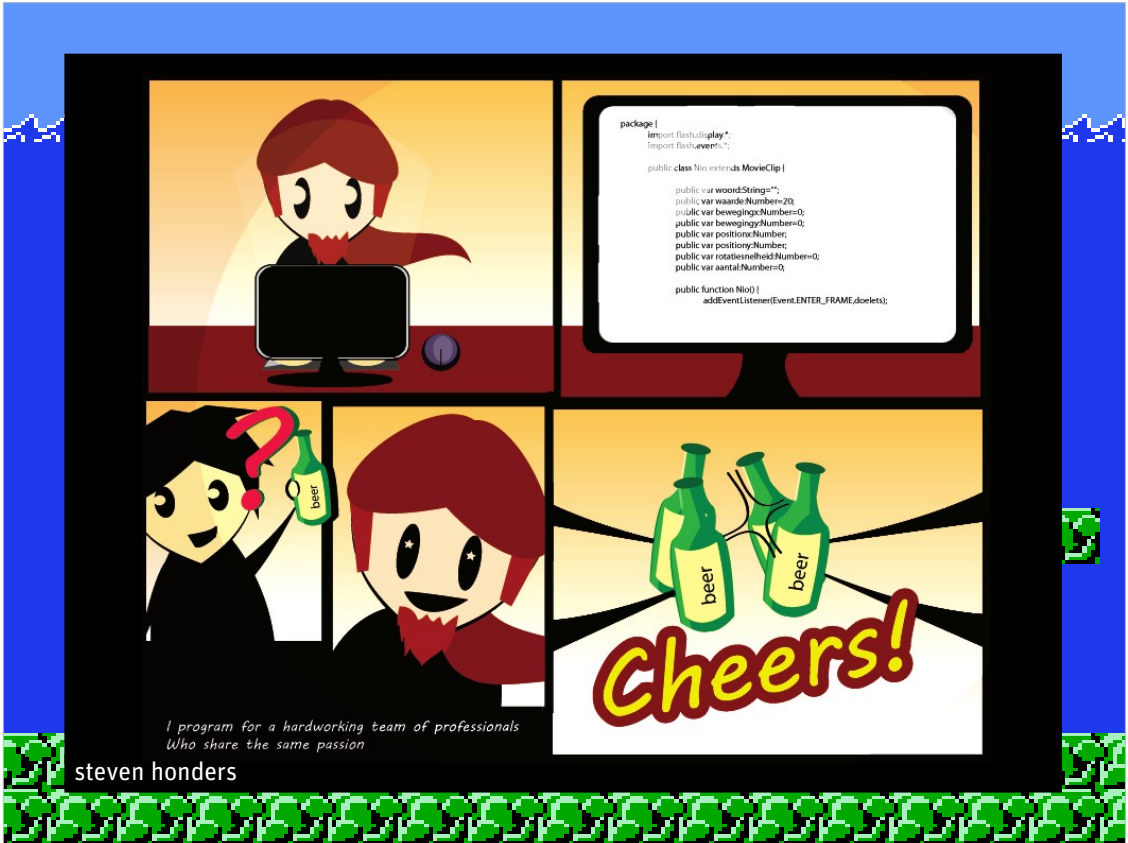
@acscherp

Deze week

- De Opdracht
- Fractals en algorithmes
- Pseudocode
- Scratch

Huiswerk review

- Bijna iedereen ingeleverd :-D
- Leuk om te bewaren en later terug te zien
- Iedereen zelfde beoordeling



```
package {
    import flash.display.*;
    import flash.events.*;

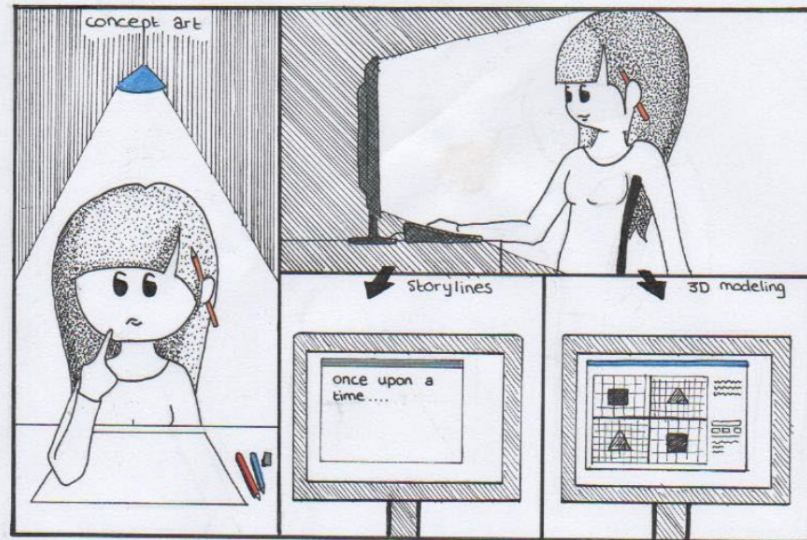
    public class Nio extends MovieClip {

        public var woord:String="";
        public var waarde:Number=20;
        public var beweging:Number=0;
        public var positie:Number;
        public var rotatiesnelheid:Number=0;
        public var aantal:Number=1;

        public function Nio() {
            addEventListener(Event.ENTER_FRAME,doelets);
        }
    }
}
```

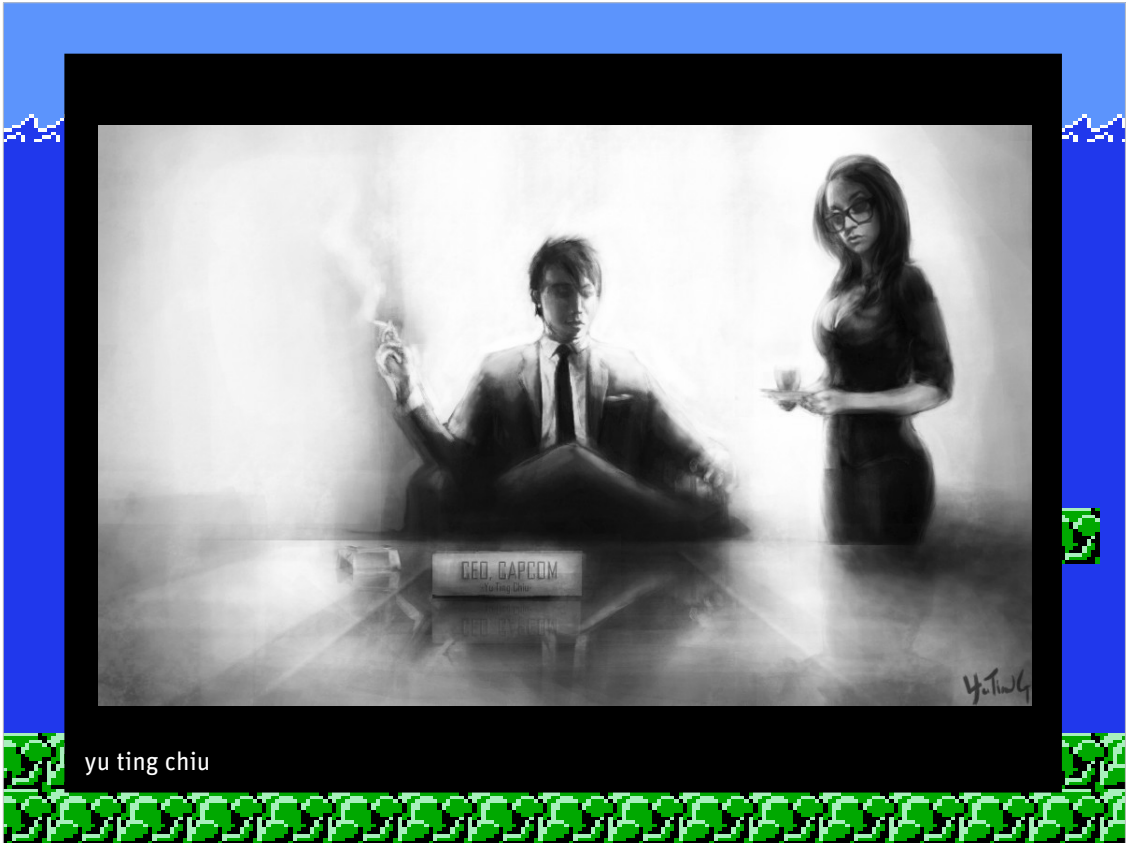
*I program for a hardworking team of professionals
Who share the same passion*

steven honders

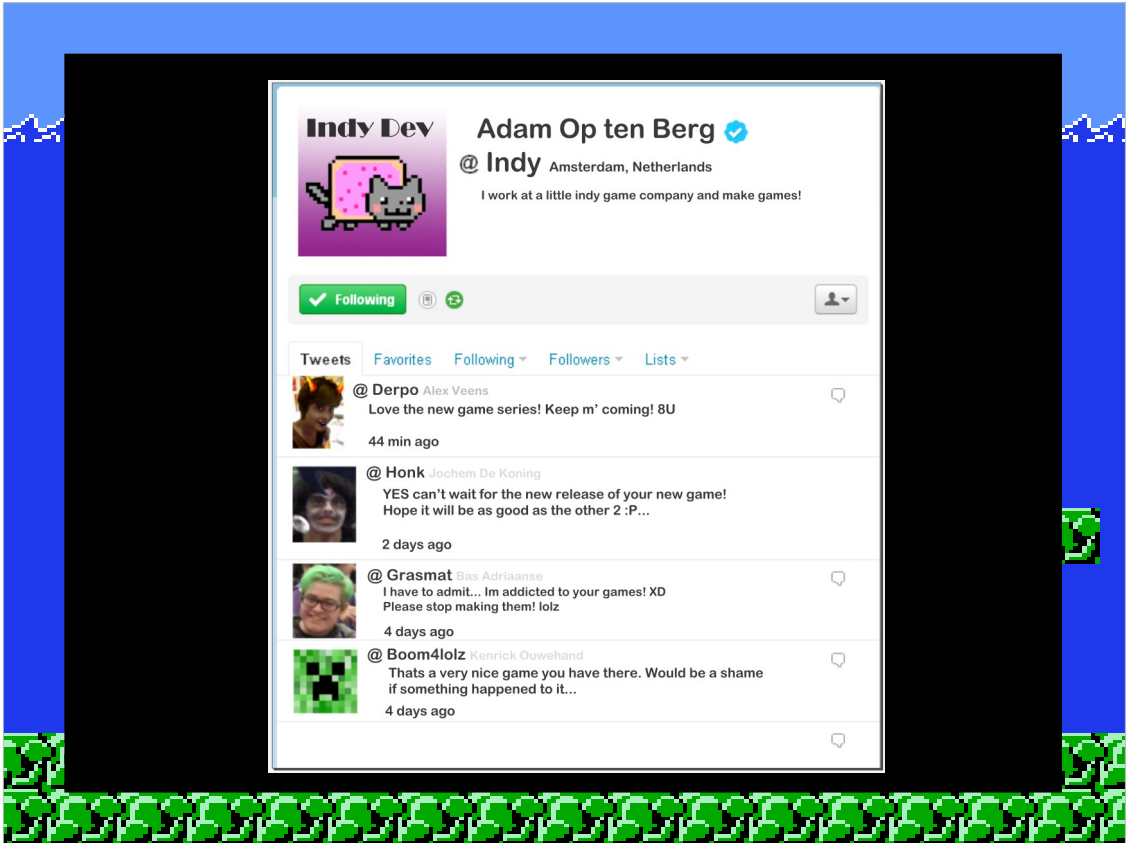


Tk zie mijzelf later, werkend bij een klein bedrijfje dat
bijvoorbeeld indiegames produceerd. Hierbij is mijn rol
het tot stand brengen van een concept en het
uitwerken ervan. Hopelijk heeft deze rol
te maken met het opzetten van storylines
en characterdesign. -Denise 😊

denise valstar



yu ting chiu



Indy Dev



Adam Op ten Berg

@Indy Amsterdam, Netherlands

I work at a little indy game company and make games!

Following



Tweets Favorites Following Followers Lists



@Derpo Alex Veens

Love the new game series! Keep m' coming! 8U

44 min ago



@Honk Jochem De Koning

YES can't wait for the new release of your new game!
Hope it will be as good as the other 2 :P...

2 days ago



@Grasmat Bas Adriaanse

I have to admit... Im addicted to your games! XD
Please stop making them! lolz

4 days ago



@Boom4lolz Kenrick Ouwehand

Thats a very nice game you have there. Would be a shame
if something happened to it...

4 days ago

- Wereld veranderen thomas versteeg
- Mannetje van alles julien ranzijn

```
ixion:~> php -a  
Interactive shell
```

```
php > $x = base64_decode('SWsgd2lsIGxhdGVyLCBhbHMgaWsgZ3Jvb3QgYmVuLCBncmFhZyBwcm9ncmFtbWV1ciB3b3JkZW4uIEhldCBsaWVmc3QgYmlqIGVlbiBrbGVpbmUgaw5kaWUtZGV2ZWxvcGVyLiBPb2sgem91IGlrIGVyZyBncmFhZyBpb2taWpuIHZyaWplIHRpamQgZWlnZW4sIGtsZWluZSBzcGVsbGVuIHdpcGxlbiBtYWtlbiEgSWsgZ2EgbnUgYmVnaW5uZW4gbWV0IGhldCBsZXJlbiB2YW4gTFVBI0gZGl0IGdhIGlrIG9vayBkb2VuIHZvb3RlbiBtYWtlbiBkYU0EgLSB1biBkYW4gd2lsIGlrIGRhYXJlYSBkZS  
BvdmVyc3RhcCBnYW5uIG1ha2VuIG5hYXJlIG9yLiBPb2sgd2lsIGlrIGVlbiBkYU0EgLSB1biBkYW4gd2lsIGlrIGRhYXJlYSBkZS  
nQsIG1hYXJlIGZGF0IGhldWZ0IGdlZW4gcHJpb3JpdGVpdC4='');  
php > echo $x;
```

```
Ik wil later, als ik groot ben, graag programmeur worden. Het liefst bij een kleine indie-developer  
. Ook zou ik erg graag in mijn vrije tijd eigen, kleine spellen willen maken! Ik ga nu beginnen met  
het leren van LUA - dit ga ik ook doen voor mijn ISA - en dan wil ik daarna de overstap gaan maken  
naar C++. Ook wil ik een beetje goed worden in pixelart, maar dat heeft geen prioriteit.
```

```
php > █
```

peter dijkstra

Literatuurlijst

- Software studies (*Matthew Fuller*)
 - o.a. artikelen “CVS”, “Variable”
- Code Complete
(*2nd Ed*) by *Steve McConnell*
- Software engineering (*Ian Sommerville*)

Deze lijst is optioneel, en bevatten geen tentamenstof.
Maar het zijn wel leuke werken om door te bladeren.

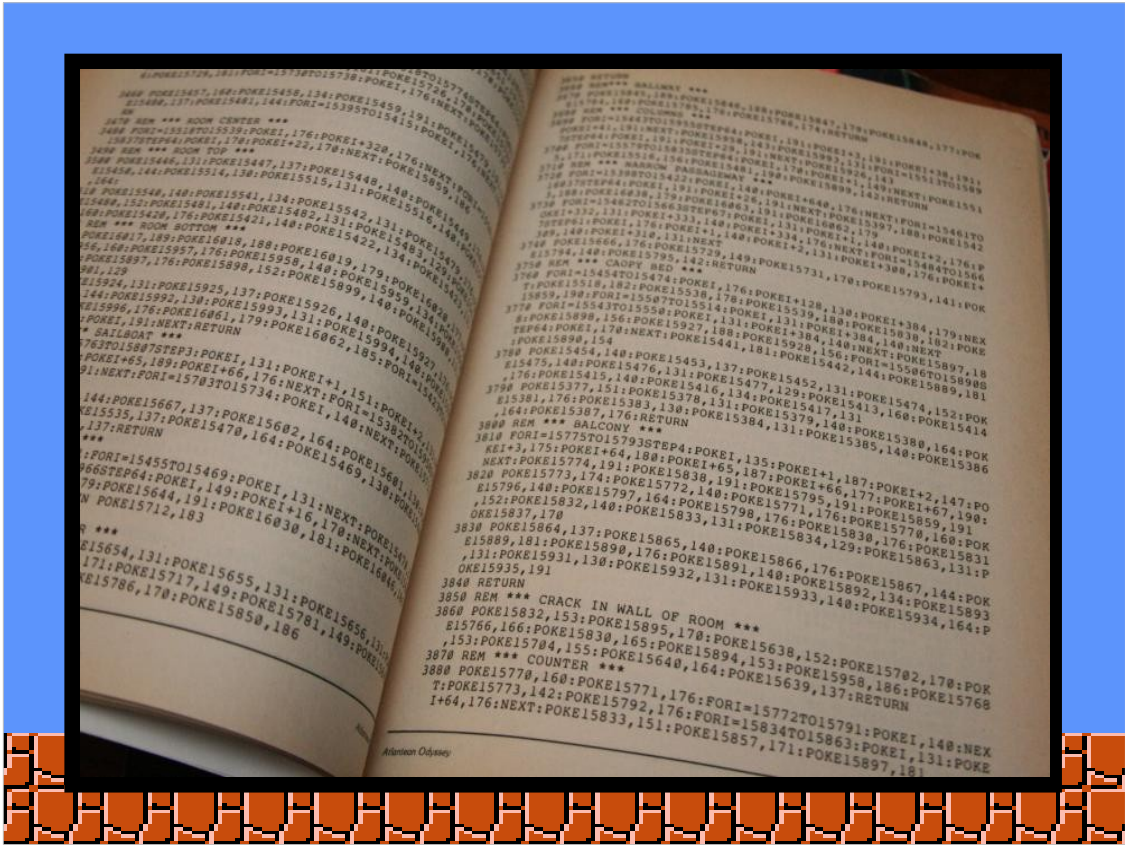
“Software studies” is een verzameling artikelen over het schrijven en de werking software vanuit een cultureel / nieuwe media standpunt.

Code Complete is *het* standaardwerk voor alle programmeurs: je wordt er een betere programmeur van. Van harte aan te raden!

Software Engineering gaat over het complete productieproces van software: van specificaties tot quality assurance.

“Methods of Development”...?

- == *thinking* of development



MoD gaat niet over implementatie maar over ontwikkelgericht denken

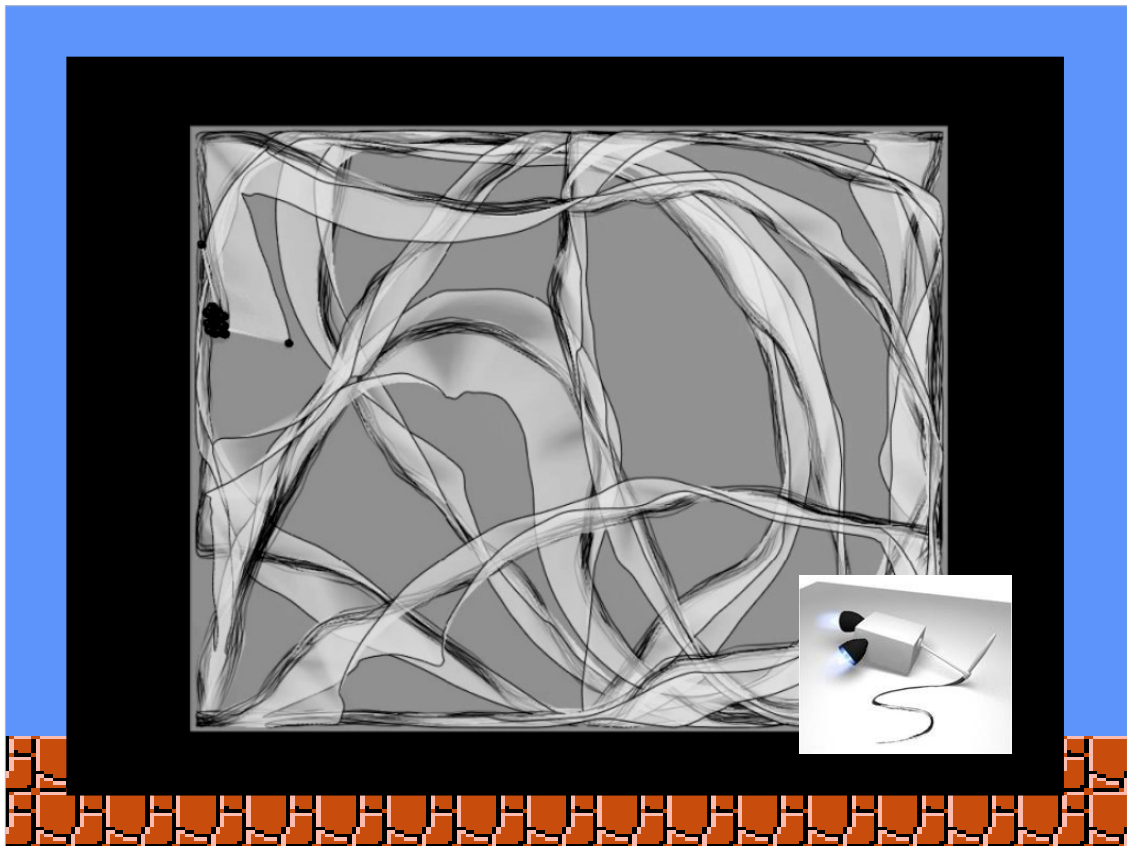


Denken over het ontwikkelproces

De programmeur als vormgever

Het is niet zo dat programmeren niets te maken heeft met vormgeven.

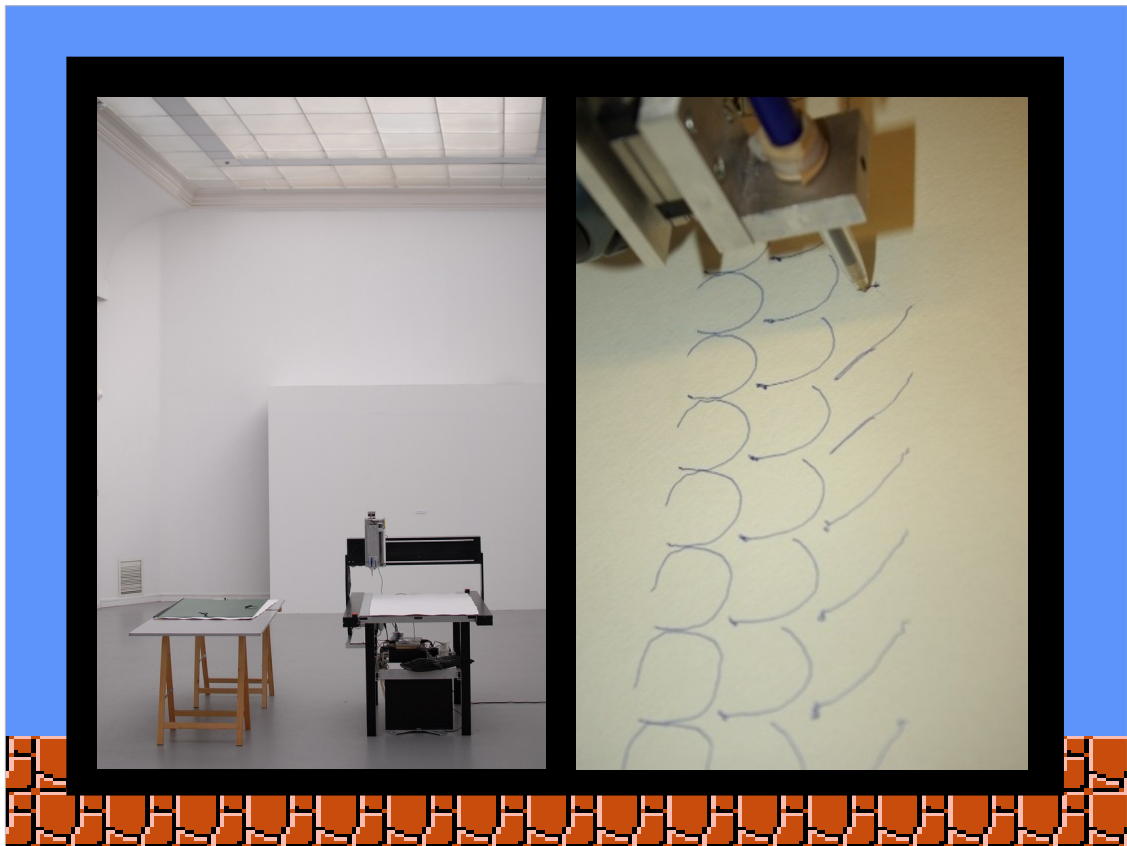
Er zijn een hoop mensen die het combineren, en er is een aparte klasse van computerprogramma's die dit doet.



Jochem van der Spek
Programmeur/kunstenaar, programmeert tekenmachines

Als programmeur stel je een aantal **condities** waar je programma zich naar gedraagt. Een stel regels

Zelfs een heel simpel aantal regels kan al een heel complex resultaat opleveren!

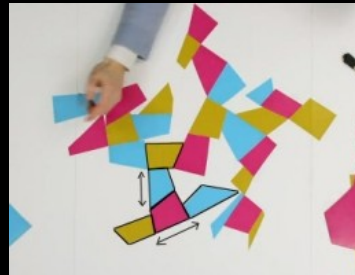


Drawing Robot #1
Concept by Jochem van der Spek
Programming by yours truly :-)

<http://miraclethings.nl/project/522/drawing-robot-1>

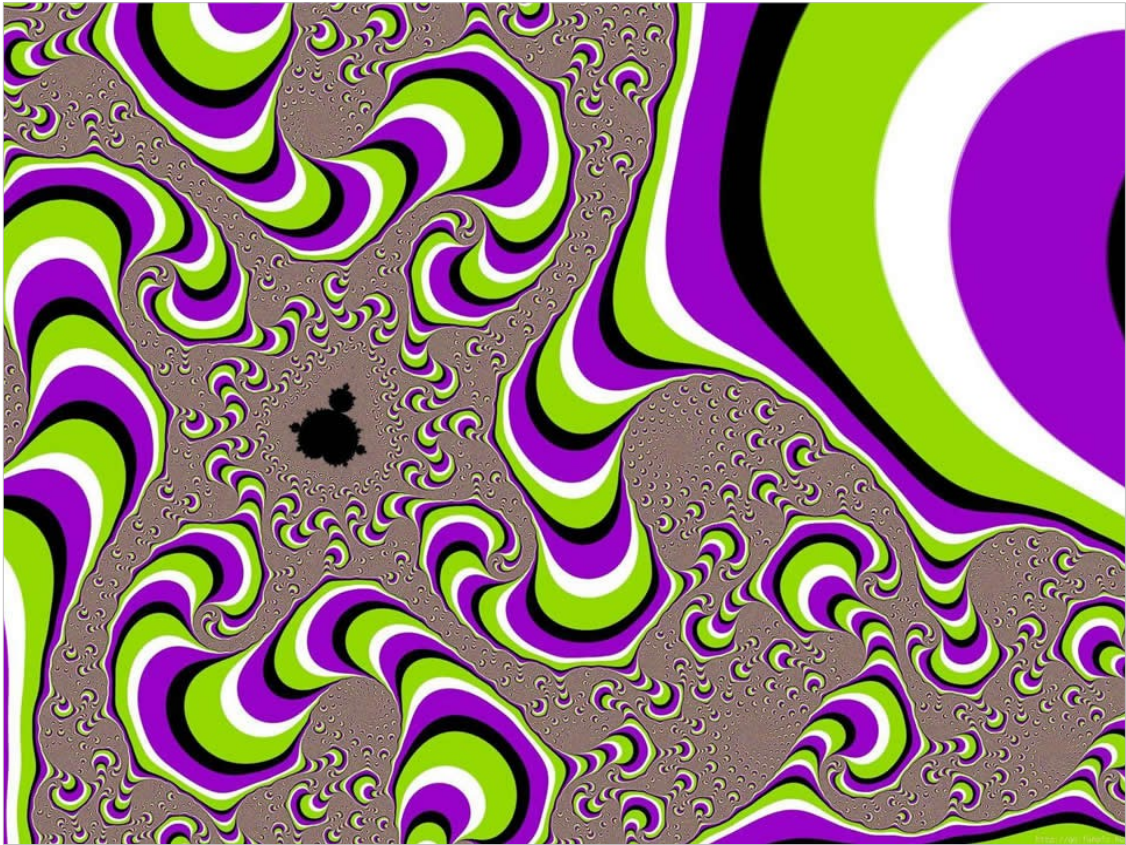
“Conditional design”

- Art initiative
- Graphic design through *process* and *logic*.
- Broccoli, Convex city plan



Een stel simpele regels, stelselmatig uitgevoerd.
Geeft een complex resultaat, dat niet altijd te voorspellen is (kan een toevalsfactor bevatten)

Dit kan ook offline, zoals het initiatief “Conditional design” hier laat zien.
<http://conditionaldesign.org/manifesto/>



Bekende procedurele graphics zijn de zgn. fractals

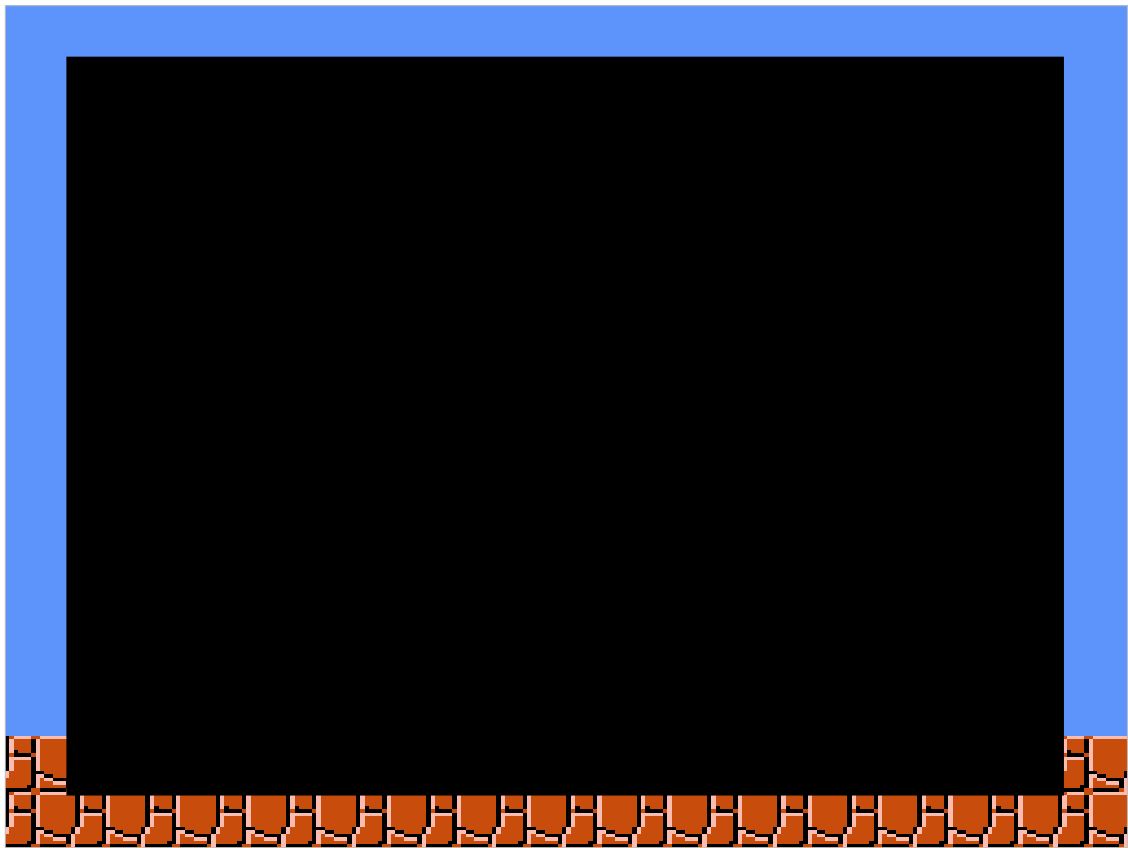
Computer-gegenereerd

- Fractals
 - Visuele uitingen van logische regels
 - Procedural textures, meshes
- Genetisch / evolutionair
 - Simulatie van de natuur: simpele regels, complex systeem
 - Games zoals *spore*

**“Genetica is the tool to use
when you absolutely must
kick booty.”**

*Kay Vriend
Texture Artist*





Programmeren

- Het aansturen van een computer via instructies
- Deze instructies vormen de *broncode* van het programma
- Instructies meestal worden niet direct in *machinetaal* geschreven
- *Compilers* en *interpreters* maken van broncode machinetaal

Tussen hetgeen je programmeert (je broncode) en hetgeen wordt uitgevoerd (machinetaal) zitten 1 of meerdere lagen van abstractie. Dat zijn andere programmeertalen die jouw code omschrijven naar iets dat de computer snapt.

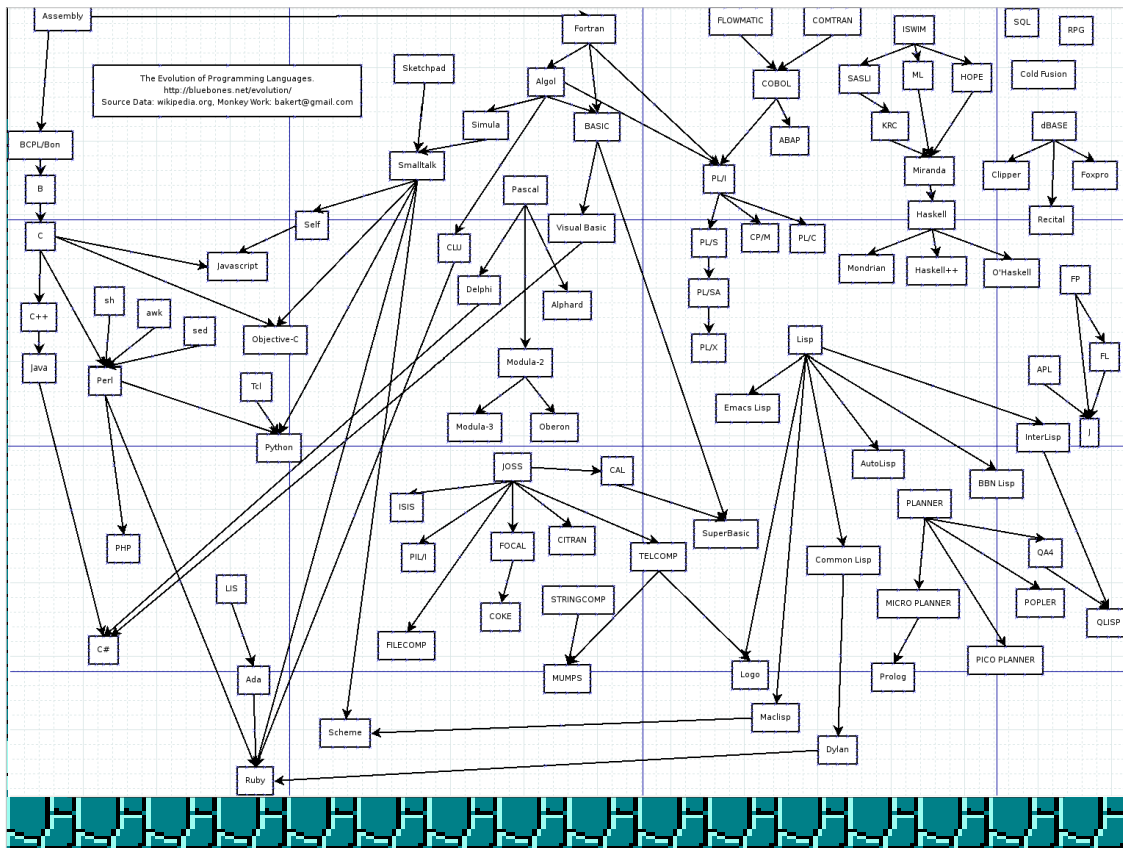
Gelukkig zijn deze tools er en hoef je je vaak geen zorgen te maken over wat er nu allemaal precies gebeurt.

Programmeertalen

- Er zijn vele duizenden programmeertalen
 - Bekende: Python, Ruby, PHP, C, C++, ActionScript, ...
 - Exotische: Shakespeare, BrainFuck, Chef, ...
- Instructies worden achter elkaar (sequentieel) uitgevoerd

Artikel over weird languages:

<http://users.soe.ucsc.edu/~michaelm/publications/mateas-software-studies-2008.pdf>
(leuk om te lezen)



<http://authenticsociety.com/Images/ComputerLanguages/EvolutionOfComputerlanguage>

http://oreilly.com/news/languageposter_0504.html

http://en.wikipedia.org/wiki/History_of_programming_languages



Maar hoe ziet code er dan uit?

Erlang

```
%%=====
%% API
%%=====

%% @spec start_link(Args) -> {ok,Pid} | ignore | {error,Error}
%% @doc Starts the server
start_link(Args) when is_list(Args) ->
    gen_server:start_link(?MODULE, Args, []).

reload(Context) ->
    z_notifier:notify(development_reload, Context).

make(Context) ->
    z_notifier:notify(development_make, Context).

%% @doc Stream specific debug information to an area on the current page.
debug_stream(TargetId, What, Context) ->
    z_notifier:notify1(#debug_stream{target=TargetId, what=What}, Context).

%% @doc Stream all debug information of a certain kind to the target id on the user agent.
observe_debug_stream(#debug_stream{target=TargetId, what=What}, Context) ->
    start_debug_stream(TargetId, What, Context).

pid_observe_development_reload(Pid, development_reload, _Context) ->
    gen_server:cast(Pid, development_reload).

pid_observe_development_make(Pid, development_make, _Context) ->
    gen_server:cast(Pid, development_make).

%% @doc Called when a file is changed on disk. Decides what to do.
%% @spec file_changed(string()) -> ok
file_changed(Verb, F) ->
    case file_blacklisted(F) of
    true -> nop;
    false ->
        case handle_file(Verb, filename:extension(F), F) of
        undefined -> ok;
        Message -> send_message(os:type(), z_string:trim(Message))
        end
    end
end,
ok.
```

Javascript

```
$ .widget("ui.inlineLightbox",
{
  _init: function() {
    var self = this;

    self.parentElement = self.element.parents(self.options.parent).first();

    self.offset = self.parentElement.offset();
    self.width = self.parentElement.width();

    self.img = $("img:first", self.element);
    self.aspect = self.img.width() / self.img.height();

    self.element.click(function(e) {
      self.clicked(e);
    });

    if (self.element.attr("id")) {
      if (document.location.hash == "#" + self.element.attr("id")) {
        self.clicked();
      }
    }
    self.zoomIcon = $("<img>").attr("src", "/lib/images/icon_zoom.png").hide();

    self.element.mouseover(function() {
      var o = self.img.offset();

      var po = self.parentElement.offset();
      var pad = (self.img.outerWidth()-self.img.width())/2;
      self.zoomIcon
        .appendTo(self.parentElement)
        .css({position: "absolute", top: o.top-po.top+pad-1, left: self.img.outerWidth()+o.left-po.left-pad-self.zoomIcon.wid
        .show();
    })
    .mouseout(function() {
      self.zoomIcon.hide();
    });
  },

  clicked: function(e) {
    var self = this;

    if (e) {
      e.preventDefault();
    }

    if (self.element.attr("id")) {
      if (document.location.hash != "#" + self.element.attr("id")) {
        var id = self.element.attr("id");
        self.element.attr("id", "");
      }
    }
  }
});
```

...WOW.

- Gelukkig ga ik jullie niet leren code schrijven :-)
- Wat we wel doen is *pseudo-code* maken.

Pseudocode

- “Pseudocode wordt gebruikt om *algoritmen* vast te leggen op een door mensen leesbare manier met behoud van de stappen.” (wikipedia)

<http://nl.wikipedia.org/wiki/Pseudocode>

En een “**algorithme**” is een bepaalde manier om een vastgesteld doel te bereiken: een stappenplan, een manier van aanpak.

Bij het schrijven van pseudocode is het het allerbelangrijks om een goed *niveau van abstractie* te vinden: niet te gedetailleerd, maar ook niet te globaal.

Als je je pseudocode goed schrijft, kan je deze later, wanneer je echt gaat programmeren, **1-op-1 overnemen** in de comments van je echte programma

Klets koppen van Holtkamp

- Verwarm oven voor op 220°C. Roer de boter, basterdsuiker en melk door elkaar. Meng kaneel, zout, amandelen erdoor, en daarna de bloem.
- Doe het beslag in een spuitzak en spuit het in kleine bolletjes op een bakplaat
- Bak de klets koppen in slechts enkele minuten in de oven. Leg ze daarna op het aanrecht om af te koelen

Het boek *de Banketbakker* van patisserie Holtkamp uit Amsterdam is een aanrader: vol met goede recepten voor koekjes, taarten, kroketten, ...

verwarm de oven voor op 220°

REPEAT

roer ingredienten door elkaar

UNTIL egaal mengel

REPEAT

sprit bolletje op bakplaat

UNTIL bakplaat vol OF deeg op

IF er is nog deeg **THEN**

bewaar rest van deeg in koelkast

END

zet plaat in oven

wacht 3 minuten

haal plaat uit oven

Pseudocode: Statements

- “Roer de ingrediënten door elkaar”
- “Verwarm de oven voor”
- *Geparametriseerd:*
 - “Verwarm de oven voor op 220° graden”
 - “Verwarm de oven voor op X graden”

Pseudocode: Condities

- Oorzaak → gevolg: “Als dit, dan dat”
- IF iets, THEN doewat, (ELSE doewatanders)

Expressie

Statement(s)

Statement(s)

- IF oventemperatuur te hoog
THEN stop met voorverwarmen

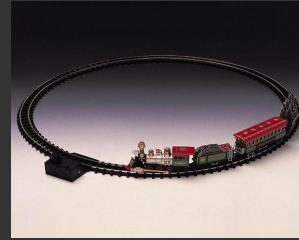


Pseudocode: Loops

- “Doe hetzelfde, totdat er iets geldig is”
- REPEAT **doewat** UNTIL **iets** (*geldig is*)

Statement(s)

Expressie



Pseudocode: Variabelen

- Een “vakje” waar iets in kan worden bewaard

SET *gewensteTemperatuur* TO 220

activeer oven

REPEAT

 SET *huidigeTemperatuur* TO de huidige oventemperatuur

UNTIL *huidigeTemperatuur* >= *gewensteTemperatuur*

Pseudocode: Expressies

- “`huidigeTemperatuur GROTER DAN gewensteTemperatuur`”
→ is een expressie
- Een “check”.
- Expressies zijn altijd “waar” (`true`) of “niet waar” (`false`)
- Worden gebruikt in condities (`IF..THEN`)
- Worden gebruikt in loops (`REPEAT..UNTIL`)

Pseudocode richtlijnen

- Duidelijke namen van dingen
- Syntax niet zo belangrijk
 - ...maar wees wel *consistent!*
- Balans abstractie / specificiteit

Balletje-balletje, te simpel:

Initialize some stuff

REPEAT

Get user input

UNTIL user choose the right cup

Show victory screen

Balletje-balletje, te complex:

```
BOOLEAN Match
INTEGER UserCup
INTEGER RandomCup
```

```
PRINT "GUESS THE CUP (C) 2011 HKU PRODUCTIONS, ALL RIGHTS RESERVED"
```

```
PRINT "Which cup do you think the ball is under?"
```

```
PRINT "Press 1 for cup number one"
```

```
PRINT "Press 2 for cup number two"
```

```
PRINT "Press 3 for cup number three"
```

```
SET NumberOfCups to 3
```

```
CALL RandomNumberGenerator with NumberOfCups RETURNING RandomCup
REPEAT
```

```
IF KeyPressed EQUALS 1 THEN
```

```
  SET UserCup to 1
```

```
ELSE IF KeyPressed EQUALS 2 THEN
```

```
  SET UserCup to 2
```

```
ELSE IF KeyPressed EQUALS 3 THEN
```

```
  SET UserCup to 3
```

```
SET Match to UserCup EQUALS RandomCup
```

```
IF Match NOT EQUALS true
```

```
  PRINT "The ball was not under the cup of your choice, please try again"
```

```
UNTIL Match EQUALS true
```

```
PRINT "Congratulations! You have guessed the right cup!"
```

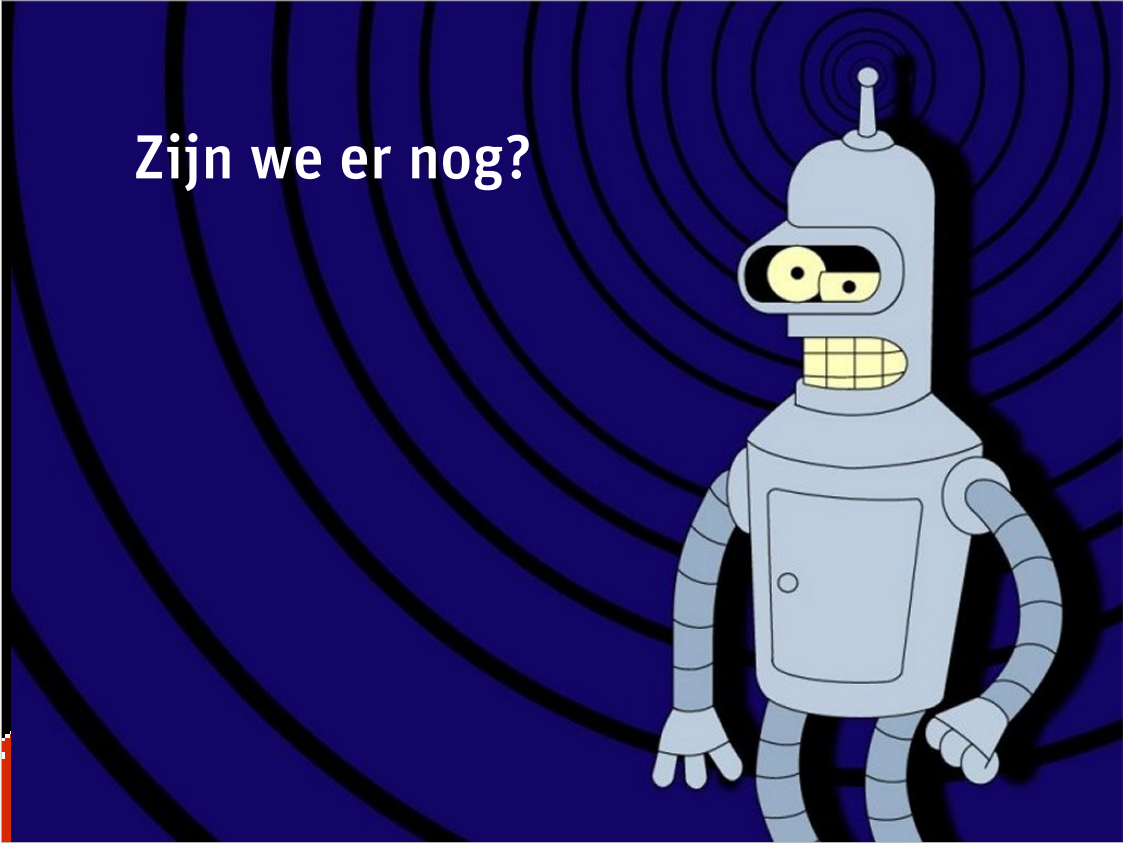
```
PRINT "Please play again!"
```

Oefenopdracht:

- Ga een dagelijkse bezigheid van jezelf na, en schrijf deze op in pseudocode. Probeer het juiste niveau van detail te vinden; verlies jezelf niet in details, maar blijf ook niet op een te hoog niveau. Verzin bezigheid waarbij je minimaal 1 variabele, 1 loop en 1 if-then-else nodig hebt.

- Deze opdracht is niet verplicht, maar het tentamen bevat in ieder geval 1 zo'n soortgelijke vraag.
- Als je je uitwerking naar me mailt krijg je feedback. Zet wel de pseudocode IN de mail (dus geen losse pdf!) dan kan ik makkelijk reply'en.

Zijn we er nog?



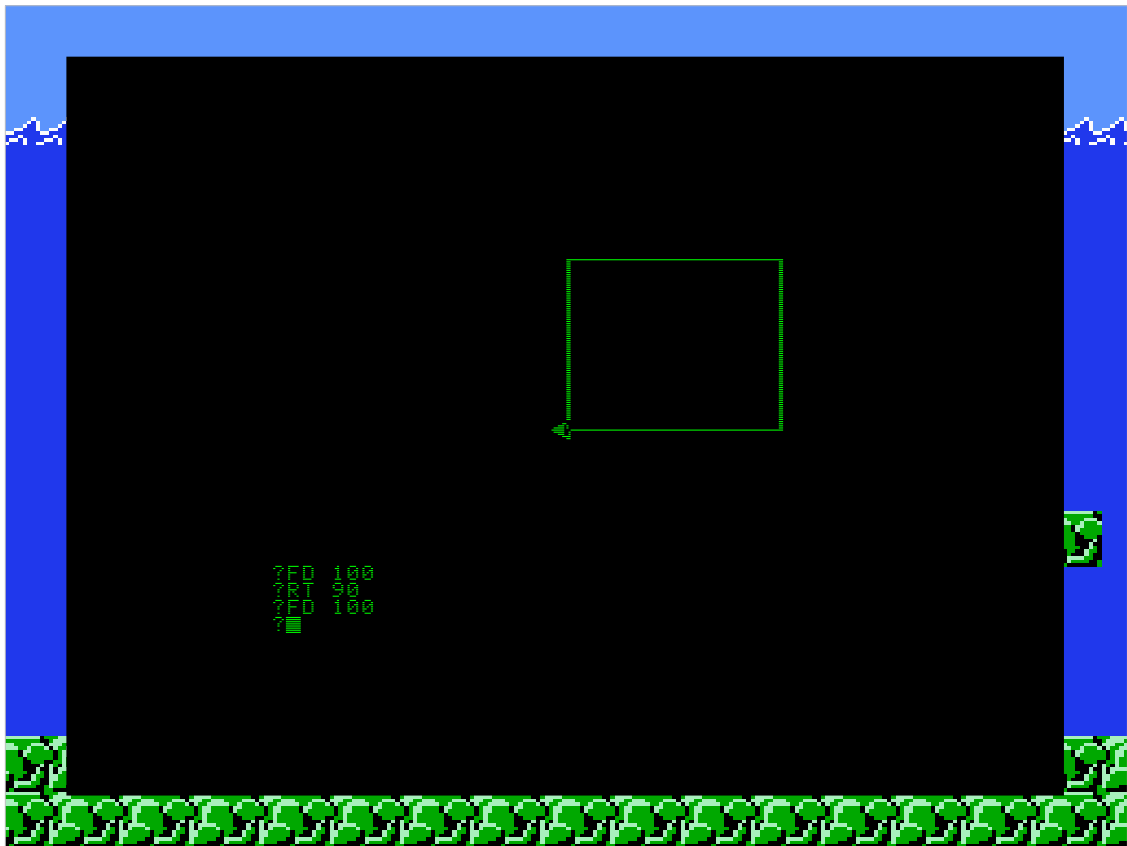
Gelukkig zijn er tools verkrijgbaar waarmee je op een toegankelijke manier kunt programmeren!



Mijn voorganger gebruikte LOGO om pseudocode uit te leggen.
(wel een iets nieuwere versie dan deze hoor)

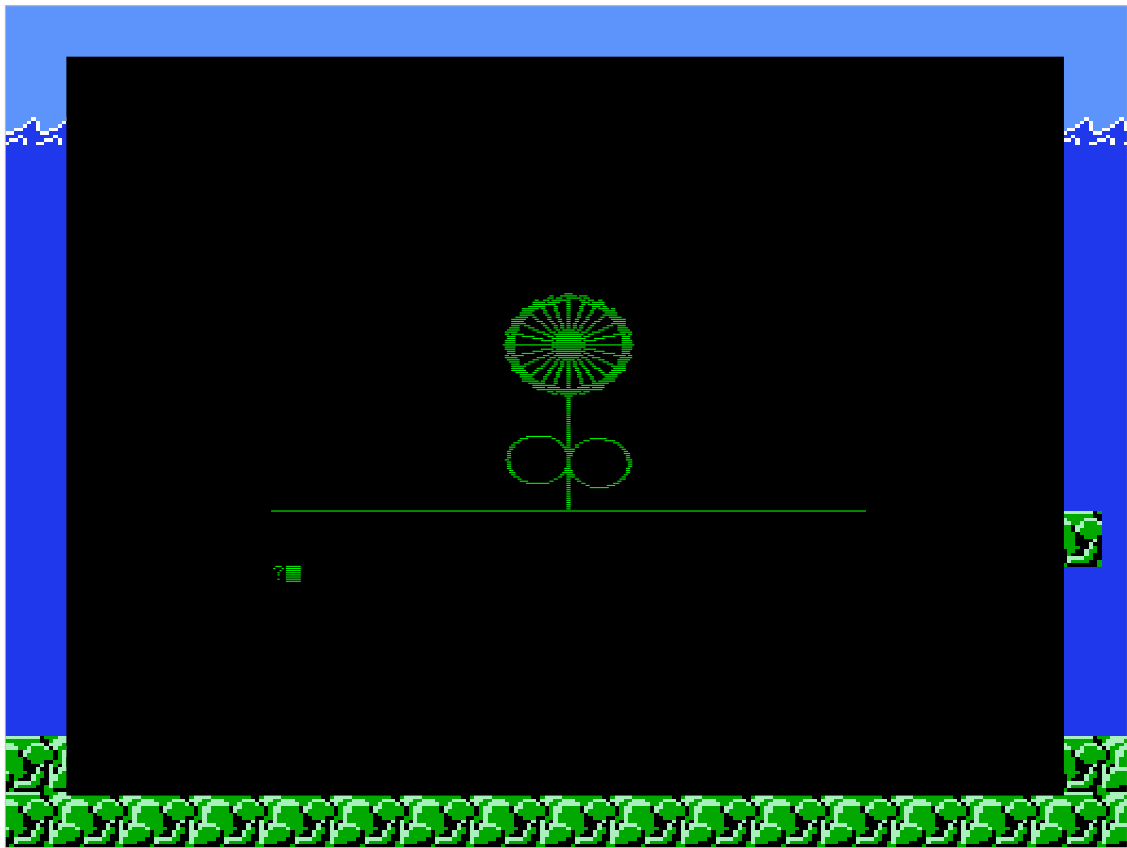


Met een "cursor" die de "Turtle" wordt genoemd, kan je tekeningen maken op het scherm.



Bijvoorbeeld een blokje

“forward 100” (stapjes),
“turn right 90” (90 graden)
En dat vier keer.

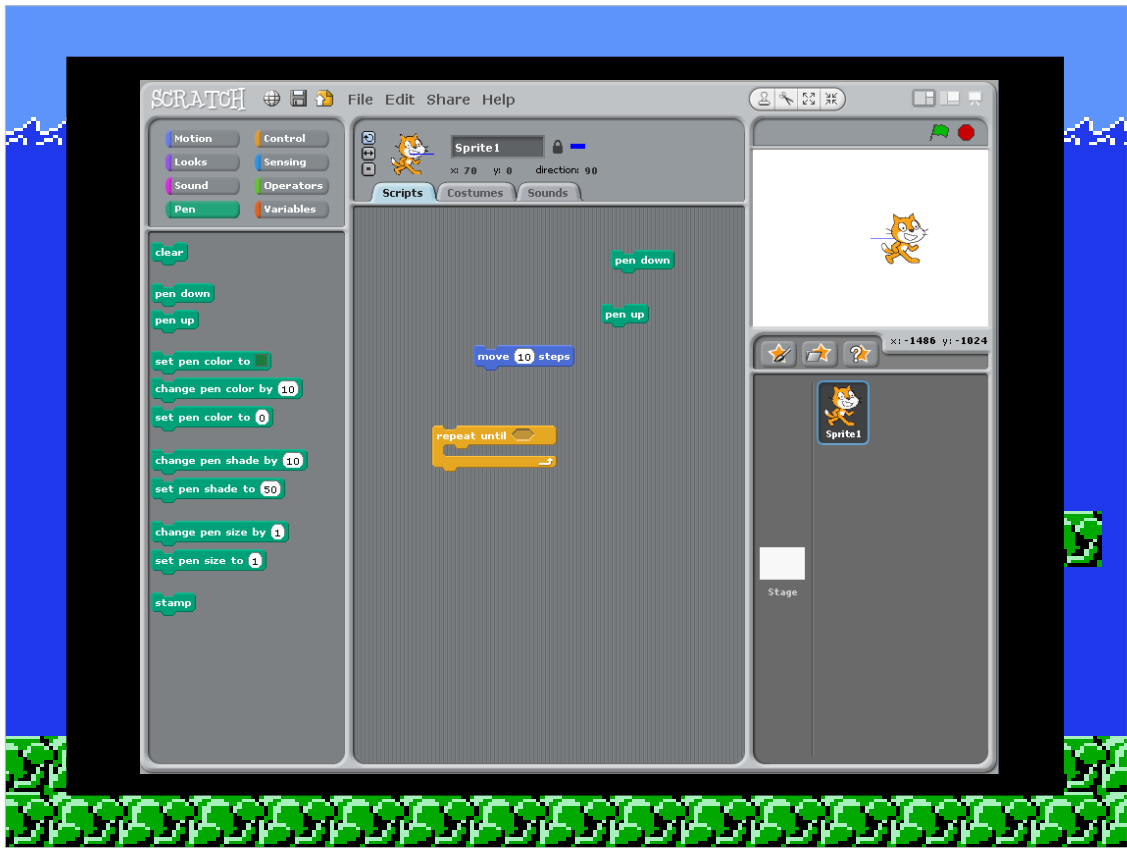


Vet complexe dingen kan je maken zo



Maar gelukkig is er nu **Scratch**

Een grafische omgeving waarin je 2-dimensionale dingen kunt programmeren met graphics, geluid en animatie.



<http://scratch.mit.edu>

Komende colleges

- ~~College 1: waar hebben we het over~~
- ~~College 2: imperatief programmeren~~
- College 3: object-orientatie
- College 4: vervolg OO, eindopdracht
- Tentamen (begin december)

Opdracht:

- Experimenteer met *Scratch* om een schetsopzet te maken voor een kleine (arcade) game. Het hoeft niet compleet af maar het moet wel iets doen! Het gaat om het script: graphics/sound zijn leuk maar tellen niet zwaar mee.
- Jullie hebben hier 2 weken voor.
- **Deadline: dinsdag 22 november 23:59u**
- arjan.scherpenisse@kmt.hku.nl

Het gaat erom dat ik zie dat je de principes snapt. Je hoeft geen hele game te maken. Maar ik ben geen fan van oranje katten dus je mag er best iets leuks van bakken.

Publiceren: upload het .sb bestand up je website en stuur mij een link.

Je mag het ook uploaden naar scratch.mit.edu, maar je **moet** me de link naar het .sb bestand sturen! (anders kan ik het niet nakijken)

Als je me het voor volgende week (do. 17 nov) al mailt dan laat ik je inzending misschien klassikaal zien volgende college.

Have fun!

Tot volgende week!

