

# WEEKENDSCHOOL LESCYCLUS

## PROGRAMMEREN –

### BEGELEIDERSINSTRUCTIE LES 1B

#### 1. INTRODUCTIE

Dit document geeft een overzicht van les 1B van de lescyclus programmeren. Afhankelijk van de Weekendschoollocatie en het aantal lessen wordt dit lesonderdeel voorafgegaan door les 1A: het instrueren van een robot om een boterham met hagelslag te smeren.

#### 2. ONBEKEND MET SCRATCH?

Als je nog nooit met Scratch gewerkt hebt, dan moet je voor de les even Scratch leren. Als je al ervaring hebt met programmeren in een andere taal dan zul je in een half uurtje wel klaar zijn. Als programmeren nieuw is dan zal het zeker een paar uur kosten.

##### 2.1 HULP VOOR BEGELEIDERS

Als je ergens niet uitkomt, schroom dan niet om voor de les te bellen met Hans de Jong, 06 204 234 84. Kan in het algemeen elke dag tot 23 uur. Op de dag zelf kun je nog wel wat vragen, maar er is heel weinig tijd en daarom is het verstandig om alle vragen tevoren beantwoord te krijgen.

##### 2.2 SCRATCH LEREN

Begin hier: <https://scratch.mit.edu/> en kies *Probeer het* of *Bekijk voorbeeld*. Dan programmeer je in Scratch 2.0. In de les programmeren we ook in Scratch 2.0 maar dan op een Raspberry Pi. Daar zijn wat fouten waar we omheen werken, maar verder is het hetzelfde.

Introducties over Scratch vind je hier: <https://scratch.mit.edu/help/>

Je kunt niet precies dezelfde geluiden toevoegen etc., maar dat zou geen belemmering moeten zijn. Als je dat wel wilt, vraag dan even om de geluidsbestanden.

In de les gebruiken we een Raspberry Pi als computer. Die werkt bijna net zoals een Mac of Windows PC. Daar hoeft je je niet op voor te bereiden.

Mocht je zelf een Raspberry Pi 2B of 3B hebben dan kan ik je ook een image voor een micro SD kaart sturen. Als je wilt kun je ook een complete setup met Raspberry Pi, monitor, toetsenbord etc. lenen.

### 3. VOOR DE LES

#### 3.1 OPBOUW VAN DE WERKPLEKKEN

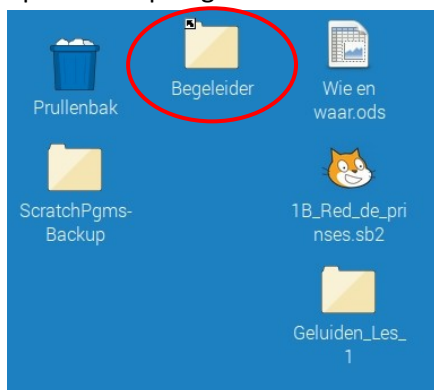
We proberen het zo te plannen dat er per begeleider maximaal 8 leerlingen zijn. Die zitten op 4 werkplekken. We proberen meer begeleiders te hebben. Elke begeleider bouwt dus maximaal 4 werkplekken op. Er is een aparte handleiding hoe dat moet.

#### 3.2 KLAAR ZETTEN VAN DE RASPBERRY PI

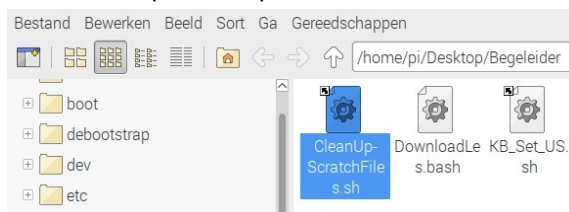
Als de werkplek klaar is, start dan de Raspberry Pi op. Daarna moeten er nog wat handelingen gedaan worden.

##### 3.2.1 START CLEANUP-SCRATCHFILES.SH

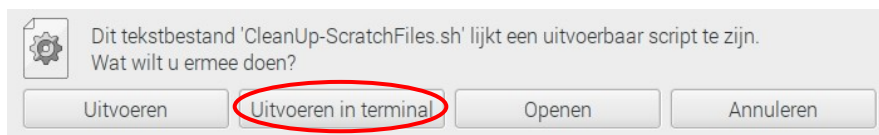
1. Open de map Begeleider



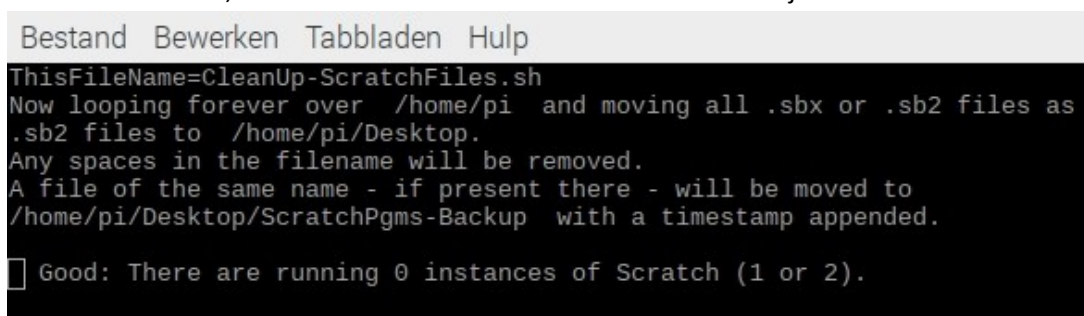
2. Dubbelklik op CleanUp-ScratchFiles.sh



3. Kies Uitvoeren in terminal



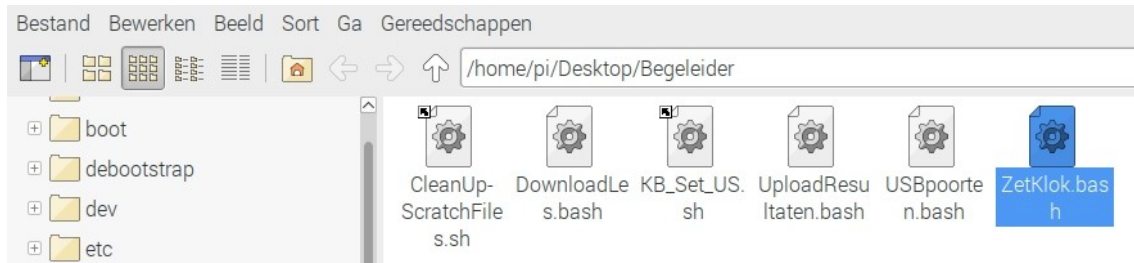
4. Je ziet nu dit venster, maar het kan ook zijn dat het venster vanzelf geminimaliseerd wordt. Als het zichtbaar is, minimaliseer het dan. Het moet de hele les blijven draaien.



Dit programma zorgt dat als je een .sb2 of .sbx file opslaat in /home/pi deze weer op het bureaublad wordt gezet, met de eventuele spaties verwijderd uit de filenaam. Dit is nodig wegens fouten in Scratch 2 op Raspberry Pi.

### 3.2.2 ZET DATUM EN TIJD

#### 1. Dubbelklik ZetKlokEnSysteem



#### 2. Kies Uitvoeren in terminal



## 4. LES 1B

- We beginnen met een introductie over hoe ze moeten computeren zonder gezondheidsklachten te krijgen.
- Daarna spelen we stukjes video af  
<http://www.hetklokhuis.nl/tv-uitzending/1142/Gezond%20computeren>  
Van 4:50 tot 8:35 en daarna van 11:07 tot 12:23.
- Dan laten we een filmpje zien over wat ze deze les gaan maken. Dit is de Scratch 1.4 versie. De Scratch 2.0 versie moet nog opgenomen worden.  
<https://www.youtube.com/watch?v=7DneCSorwug>
- We leggen we uit dat programmeren de één na laatste stap is. De laatste stap is testen. Maar het begint met het analyseren van het probleem en bedenken hoe je het aanpakt. Dat gaan we daarom eerst doen. Daarvoor gaan ze in groepjes (een aantal tweetallen) met een begeleider zitten en praten over welke poppetjes er zijn en wat die doen.
- Daarna doen we de opgave  
*Een doolhofspel – red de prinses*
- Afhankelijk van hoeveel eigen werk ze dan al gedaan hebben laten we ze aan het eind van het eerste lesblok rondlopen om bij elkaar te kijken en inspiratie op te doen voor het tweede lesblok, waarbij ze doorgaan met deze les 1B.

De les is een variatie op de les van <http://www.codeuur.nl/lesmateriaal#smartlife>. Die zijn voor de leeftijd 8 – 12 jaar om in 60 – 90 minuten te doen. Op de Weekendschool zijn de leerlingen 11 of 12 jaar en we hebben 2x75 minuten. Naar aanleiding van de eerste keer hebben we besloten om de les uit te breiden met extra concepten en niet een van de andere lessen te nemen. Deze les maakte heel wat creativiteit los en het leek beter om ze daarmee door te laten gaan.

#### 4.1 WELK LEERPROCES?

De Lifelong Kindergarten group van het MIT in Boston, waar Scratch ontwikkeld is, propageert het Creative Learning: leren door te doen. Zij leggen de nadruk op het door de leerlingen zelf laten ontdekken en spelen en zo te leren. Ik geloof daar ook zeer in, echter we hebben maar drie zondagen en willen ze toch wat meer meegeven. Daarom is gekozen om ze opdrachten te laten maken. Voor spelen en eigen inbreng is gelegenheid aan het eind. Maar als ze eerder zelf dingen willen uitproberen en een wat andere richting in gaan dan is dat prima, maar kijk wel dat ze niet in een onderwerp blijven hangen omdat ze de rest (zogenaamd) niet snappen.

#### 4.2 LESDOEL VAN LESDAG 1

- Leren hoe ze zonder gezondheidsklachten met beeldscherm, muis en toetsenbord kunnen werken.
- Laten kennismaken met het analyseren van een probleem.
- Eerste kennismaking met programmeren in Scratch.
- Laten zien dat je met heel weinig moeite een animatie kunt maken.
- Ze kennis laten maken met de bouwstenen van Scratch
  - o Herhalen
  - o Testen op gebeurtenissen (toetsaanslagen, het raken van een object)
  - o Beweging
  - o Variabelen
  - o Geluid
  - o Random getallen
  - o Zenden en ontvangen van berichten tussen de sprites
- Ze laten tekenen

## 5. ALGEMENE AANWIJZINGEN

### 5.1 KOPIËREN TUSSEN DE SPRITES

Je kunt opdrachten kopiëren van de ene naar de andere sprite door de opdracht of het blok opdrachten op te pakken en naar de betreffende sprite te slepen. Je zult zien dat er een lijntje om de sprite komt als die geselecteerd is. Het is het handigste om het te kopiëren blok links boven op te pakken.

Je kunt geen opdrachten naar een nieuwe sprite *verplaatsen*. Je moet kopiëren en dan de originele opdrachten verwijderen. Als je niet helemaal zeker bent dat een blok is gekopieerd, klik dan eerst even op de sprite om te zien of het echt gelukt is alvorens het oude blok te verwijderen.

### 5.2 OPSLAAN

Laat ze regelmatig hun werk opslaan. Anders zijn ze het kwijt bij stroomuitval.

Scratch 2.0 op Raspberry Pi heeft een aantal fouten. Een daarvan is dat als je iets wilt opslaan het altijd per default in de /home/pi map gebeurt. Scratch vergeet waar vandaan het bestand geopend was. Ook is er een probleem als de bestandsnaam spaties heeft. Als je dan het dan weer wilt opstarten door te dubbelklikken dan gaat dat fout. Tot slot, als extension blocks worden gebruikt in les 2, wordt het programma altijd opgeslagen met .sbx extensie waar Scratch daarna bij opstarten niets mee kan.

Er is daarom een script *CleanUpScratchFiles.bash* in de map Begeleiders. Dat moet je eenmalig starten na het opstarten van de Raspberry Pi. Die verplaatst alle bestanden met .sb2 of .sbx extensie naar het bureaublad nadat de spaties uit de filenamen zijn gehaald. Als er al een bestand met dezelfde naam op het bureaublad stond dan wordt die verplaatst naar de map *BackupScratchFiles* op het bureaublad, met achter de bestandsnaam de datum en tijd. Zo is het ook altijd weer mogelijk om een vorige versie terug te vinden.

Tot slot kijkt het script hoeveel Scratch instantiaties actief zijn. Als het twee of meer zijn dan gaat de titelbalk knippen. Voor les 2 is het belangrijk dat er slechts één instantiatie actief is.

### 5.3 VOORTGANG NOTEREN

Nu we deze les nog maar een paar keer gedaan hebben en steeds ook iets hebben veranderd willen we uitzoeken hoeveel tijd ze nodig hebben om de les te doen. Zodat we in latere lessen bijvoorbeeld weten dat ze na een uur bij opdracht X moeten zijn. Daarom krijgt elke begeleider een formulier voor de groepjes die hij of zij begeleidt en vult daar op gezette tijden (bijvoorbeeld elk kwartier) in bij welke opdrachtnummer elk groepje is. Als de leerlingen dat zien, leg ze dan uit dat het geen wedstrijd is, dus dat ze gewoon moeten doorwerken, en dat het er niet om gaat wie het snelste is. We noteren geen namen van de leerlingen. Het is puur voor ons eigen inzicht in het lesverloop.

## 6. OPMERKINGEN PER ONDERDEEL

### 6.1 HET ANALYSEREN VAN DE OPDRACHT

Vraag ze wat ze gezien hebben in het filmpje. Wat voor grote stappen of grote blokken zien ze daarin. Bespreek dat met ze. Experimenten met het laten opschrijven hiervan was geen groot succes, dus dat laten we achterwege.

Grote blokken waar je aan kunt denken:

1. Een doolhof verkrijgen
2. Een poppetje (de kat) in vier richtingen laten bewegen aan de hand van toetsaanslagen.
3. Als de kat de muur raakt, keer dan terug naar *af* en zeg even *Au!*
4. Als de kat bij de prinses komt dan moet de kat iets uitroepen.
5. Het spel eindigen als de tijd over is
6. Het spel eindigen als de prinses is gered
7. Kopstoten kunnen tellen.
8. De prinses moet weten als de kat in gevaar is en dan op en neer springen.
9. Geluid laten afspelen

Vertel ze dan dat ze in de les nu een vast schema volgen, want we kunnen de instructies niet aanpassen aan de hand van wat ze hier bedenken. En ze moeten de basis stappen nog leren. Maar vraag ze waar ze denken dat je in de praktijk mee zou beginnen.

**Antwoord:** met de meest onzekere of riskante dingen, die wel essentieel zijn. Daar maak je een prototype van om te kijken of het concept werkt.

Verder is de ontwikkelmethode zo dat je iets kleins bouwt, dat gaat testen en laat zien aan de gebruikers en dan bouw je nieuwe dingen erbij. Dit is het *agile* ontwikkelmodel.

### 6.2 HET DOOLHOF IS AL AANWEZIG

Ze zouden het doolhof ook zelf kunnen tekenen, echter naar aanleiding van eerdere ervaringen hebben we het zelf tekenen naar het eind verplaatst. Dit is gedaan omdat ze zich compleet kunnen verliezen in het tekenen van het mooist denkbare doolhof – dat dan soms ook nog eens niet echt als doolhof bruikbaar is – en verdwijnt als je op Clear drukt terwijl je niet precies weet wat dat doet (alles weggooien).

Het eerst laden van een achtergrond is ook geen goede optie omdat we dan direct komen bij een andere fout in Scratch op RPi: de muisknoppen doen het niet bij import en dan moet er genavigeerd worden met de pijltjestoetsen. Geen fijne binnenkomer.

### 6.3 OPDRACHT 11: WAAROM STAAT ER EEN ZEG MET BLANKO TEKST

Er wordt gevraagd waarom er een blokje *zeg* staat met een blanco tekst. Dat is om de eerder vertoonde tekst weg te halen.

## 6.4 VARIABELEN

We laten ze de variabelen globaal aanmaken, dus beschikbaar in alle sprites. Het is in de praktijk beter om de variabelen alleen te laten gelden voor de sprite die ze nodig heeft, maar dat is een extra concept waarvan ze het nut nu niet zullen inzien.

## 6.5 SIGNALLEN

Als je bij het signaal “In gevaar” van de kat niet zou wachten na het zenden van het signaal dan worden er heel veel signalen gestuurd en dat geeft wat bijzondere effecten. Daarom wachten we na het zenden van een signaal.

## 6.6 TEKENEN VAN HET DOOLHOF

Dit onderdeel is na ervaring met eerdere lessen naar het eind verplaatst. Zoals eerder al gezegd, als je ze laat beginnen een doolhof zelf te laten tekenen dan spenderen ze daar heel veel tijd aan en komen soms met iets dat niet echt als doolhof bruikbaar is.

Als ze in verschillende kleuren tekenen, ook al lijken soms b.v. twee kleuren rood hetzelfde, betekent dat overal waar ze testen op het raken van een kleur moeten testen op twee of meer kleuren. Dat maakt het niet makkelijker. Beter om dat te vermijden.

## 7. HET COMPLETE PROGRAMMA

Hieronder staat het complete programma. Drie sprites en het speelveld.

### 7.1 SPEELVELD



## 7.2 KAT

Zorg dat de kat verschijnt als die eerder opgegeten was door de haai. Schuif de kat naar de startpositie.

x: 10  
y: 70

```

    when green flag is clicked
      show
      set Aantal kopstoten to 0
      go to x: -200 y: -140
      loop
        if key pressed: right arrow
          x = x + 5
        if key pressed: left arrow
          x = x - 5
        if key pressed: up arrow
          y = y + 5
        if key pressed: down arrow
          y = y - 5
        if I am touching color: red
          play sound: Au Au Au.mp3
          change color effect by 100
          say Au!
          Aantal kopstoten = Aantal kopstoten + 1
          go to x: -200 y: -140
          say 
          change color effect by -100
        if I am touching: Prinses
          say Joepie! Gewonnen!
          play sound: Gewonnen
          stop all
  
```

when green flag is clicked

show

set Aantal kopstoten to 0

go to x: -200 y: -140

loop

if key pressed: right arrow

x = x + 5

if key pressed: left arrow

x = x - 5

if key pressed: up arrow

y = y + 5

if key pressed: down arrow

y = y - 5

if I am touching color: red

play sound: Au Au Au.mp3

change color effect by 100

say Au!

Aantal kopstoten = Aantal kopstoten + 1

go to x: -200 y: -140

say

change color effect by -100

if I am touching: Prinses

say Joepie! Gewonnen!

play sound: Gewonnen

stop all

when green flag is clicked

loop

if distance to: Haai < 70

send signal: In gevaar and wait

when I receive signal: Opgegeten

hide



### 7.3 PRINSES



The code for the Princesses starts with a 'when green flag is clicked' event. It says 'Help!' and moves to x: 200, y: 129. A loop block contains a 'wait a random time between 1 and 5 sec.' block followed by another 'say Help! 2 sec.' block. Below the loop, there is a 'when I receive a signal: In danger' event, which moves y by 10, waits 0.1 sec, and moves y by -10.

De "ga naar ..." staat niet in de les, maar is nodig om te zorgen dat als de prinses naar boven gesprongen was op het moment dat het programma afliep ze niet omhoog blijft hangen.

Hier springt de prinses eenmaal op en neer.

x: 200  
y: 129

### 7.4 HAAI



The code for the Shark starts with a 'when green flag is clicked' event. It sets size to 50%, moves to x: 180, y: 10, and enters a loop. The loop moves 2 steps, turns at the edge, and checks 'if I touch Kat?'. If yes, it sends 'Opgeeten' signal, starts 'Game Over 1' sound, sets size to 100%, says 'Game over!', waits 3 sec, and stops all.

Zet de grootte op 50% en schuif de haai naar de beginpositie. Je weet immers niet waar die is geëindigd in het vorige rondje van het spel.

De kat moet verdwijnen, maar dat kan alleen de kat zelf doen. Die moet daarom een signaal krijgen.

Je moet wachten, omdat "stop alle" de hele uitvoering stopt en daarmee ook de "zeg Game over!" zou teniet doen. Alternatief zou je "zeg Game over! 3 sec" kunnen gebruiken.

x: 29  
y: 10

## 8. BOEKJES MEE NAAR HUIS?

Als we beginnen met les 1A en dus de volgende week doorgaan met de tweede helft van 1B dan worden alle boekjes ingenomen, zodat ze die volgende week weer kunnen gebruiken.

Het is niet de bedoeling dat ze in de boekjes schrijven.

Aan het eind van les 1B mogen ze als ze graag willen het boekje mee naar huis nemen. Maar als ze dat niet heel graag willen, neem ze dan in voor een eventueel volgend gebruik bij een volgende locatie.

## 9. VRAGEN EN ANTWOORDEN

Hier een aantal vragen die ze mogelijk stellen en de antwoorden

- Waarom heet dit ding Raspberry Pi?
  - o Het beestje moet een naam hebben.
  - o De bedenkers wilden een naam van fruit gebruiken om nostalgische redenen. Raspberry is een framboos. Vroeger werden meer computers naar fruit genoemd: Apricot (abrikoos) en je kent er vast nog wel een: Apple.
  - o Pi komt van Python, de naam van een programmeertaal die je kunt gebruiken. En als je Pi spelt als Pie dan betekent het: taart. Dus frambozentaart.
  - o Het heeft navolging gekregen, want iemand is Banana Pi gestart.
  - o En er is nu ook een Orange Pi, wat een complete computer is ala de Raspberry Pi, maar dan voor rond de 10 euro (excl. SD kaart, voeding, monitor, toetsenbord en muis).
- Wat kost een Raspberry Pi en deze opstelling?
  - o Deze Raspberry Pi, als je die in Eindhoven koopt kost (sossolutions.nl) 58,95 euro (De Raspberry Pi, behuizing, SD kaart en netvoeding. Gratis verzending.
  - o Tweede hands monitoren kun je kopen voor ca. 25 euro, degene die we in de les gebruiken waren tussen de 8,50 en 10 euro. Tweede hands muizen 1,25 euro en toetsenbord 2,50 euro. Kabel voor de monitor kost een paar euro. Maar het hangt heel erg af waar iets toevallig aangeboden wordt en er kunnen verzendkosten bijkomen.
  - o Om alles nieuw te kopen moet je naast de Raspberry Pi rekenen op ca. 120 euro (100 euro monitor, 15 euro muis en toetsenbord, 5 euro HDMI kabel).
  - o Overigens zijn er ook mogelijkheden om een heel klein scherm aan te sluiten en dan ben je minder kwijt, maar heb je ook minder resolutie.
- Zijn er nog goedkopere computers?
  - o Ja. Afgezien van de Orange Pi die ik al wel heb maar nog geen tijd voor gehad om uit te proberen zullen we in de tweede en derde les een computer gebruiken (Arduino Nano) die samen met de voeding en wat cijferdisplays en knopjes ca. 5 euro kost. Maar daar kun je dan ook veel minder mee, en je hebt een grotere computer nodig (zoals de Raspberry Pi) om die te programmeren.
- Kunnen ze thuis op een computer dit ook doen?
  - o Ja, er is een beschrijving hoe ze dat moeten doen. Als ze hun naam invullen in de spreadsheet op het bureaublad (*Wie en waar.ods*) en aangeven dat ze het willen hebben, dan krijgen ze hun werk en een blad met uitleg per email toegestuurd.
  - o Hebben ze geen computer waar ze thuis op kunnen en/of mogen werken? Laat ze het noteren. We vinden zeker een oplossing.